

การพัฒนาเกมมาสเปกโตรมิเตอร์โดยใช้เครื่องเล่นเกมส์แบบพกพา

นายปิยเทพ ชอบทำกิจ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชานิวเคลียร์เทคโนโลยี ภาควิชาวิศวกรรมนิวเคลียร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2554
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ที่ส่งผ่านทางบัณฑิตวิทยาลัย

The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository (CUIR)
are the thesis authors' files submitted through the Graduate School.

DEVELOPMENT OF A GAMMA SPECTROMETER USING A PORTABLE
GAMES PLAYER

Mr. Piyathep Chobthumkit

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Nuclear Technology

Department of Nuclear Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2011

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การพัฒนาเกมมาสปีกโตรมิตอร์โดยใช้เครื่องเล่นเกมส์
	แบบพกพา
โดย	นายปิยเทพ ชอบทำกิจ
สาขาวิชา	นิเวศศาสตร์เทคโนโลยี
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	อาจารย์เดโช ทองอร่าม
อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม	ผู้ช่วยศาสตราจารย์สุวิทย์ ปุณณชัยยะ

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้หัวข้อวิทยานิพนธ์ฉบับนี้เป็นส่วน
หนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.บุญสม เลิศหิรัญวงศ์)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รองศาสตราจารย์ สมยศ ศรีสถิตย์)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(อาจารย์เดโช ทองอร่าม)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม
(ผู้ช่วยศาสตราจารย์สุวิทย์ ปุณณชัยยะ)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์อรรถพร ภัทรสุมันต์)

..... กรรมการภายนอกมหาวิทยาลัย
(อาจารย์ ดร.บุญช่วย ศรีธรรมศักดิ์)

บิณฑุพ ขอบทำกั : การพัฒนาแกมมาสเปกโตรมิเตอร์โดยใช้เครื่องเล่นเกมส์แบบพกพา. (DEVELOPMENT OF A GAMMA SPECTROMETER USING A PORTABLE GAMES PLAYER) อ. ที่ปริกษาวิทยานิพนธ์หลัก: อ.เดโช ทองอร่าม, อ. ที่ปริกษาวิทยานิพนธ์ร่วม: ศศ. สุวิทย์ ปุณณชัยยะ 100 หน้า.

วิทยานิพนธ์นี้มีวัตถุประสงค์เพื่อพัฒนาแกมมาสเปกโตรมิเตอร์ที่ทำงานร่วมกับเครื่องเล่นเกมส์แบบพกพา โดยได้ออกแบบระบบให้สามารถเลือกโหมดการทำงานได้ทั้งโหมดวัดสเปกตรัมรังสีแกมมาพร้อมแสดงสเปกตรัมพลังงานและโหมดวัดอัตราปริมาณรังสีแบบเครื่องสำรวจรังสี ซึ่งสามารถแสดงอัตรานับรังสีในหน่วย cps และแสดงอัตราเอกซ์โพเซอร์ในย่าน 0-100 mR/h แบ่งย่านวัดย่อยเป็น 3 ช่วง คือ x1, x0.1 และ x0.01 การจัดการข้อมูลวัดรังสีอาศัยการทำงานร่วมกันระหว่างไมโครคอนโทรลเลอร์ขนาด 8 บิตและไอซีพีเอสดี (PSD) เชื่อมโยงกับเครื่องเล่นเกมส์แบบพกพา ในส่วนของระบบวัดรังสีได้ออกแบบให้ใช้งานกับหัววัดเรืองรังสีชนิด CsI(Tl) ขนาดผลึก 1 cm x 1cm ประกอบกับฟิโนโตไดโอด มีวงจรหลัก 5 ส่วน ประกอบด้วย วงจรแหล่งจ่ายไฟฟ้้าศักดาต่ำ วงจรขยายสัญญาณพัลส์ วงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดี่ยว วงจรนับจำนวนพัลส์พร้อมโปรแกรมตั้งเวลาและวงจรเชื่อมโยงสัญญาณซึ่งทำหน้าที่ถ่ายโอนข้อมูลและสัญญาณควบคุมระหว่างไอซีพีเอสดีกับเครื่องเล่นเกมส์แบบพกพา สำหรับวงจรมับจำนวนพัลส์พร้อมโปรแกรมตั้งเวลาและวงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดี่ยวทำงานภายใต้การควบคุมของไมโครคอนโทรลเลอร์

จากการพัฒนาพบว่าเครื่องเล่นเกมส์แบบพกพาช่วยลดการใช้ฮาร์ดแวร์ในส่วนแสดงผลลงและมีความยืดหยุ่นในการออกแบบเชิงกราฟฟิค ผลทดสอบการทำงานในโหมดวัดสเปกตรัมรังสีแกมมาพบว่าระบบมีความสามารถในการแจกแจงพลังงาน 12.9% ที่พลังงานรังสีแกมมา 662 keV ของ Cs-137 ส่วนในโหมดวัดอัตราปริมาณรังสีสามารถวัดอัตรานับรังสีได้สูงสุด 10 kcps และวัดอัตราเอกซ์โพเซอร์ที่ย่านวัด x1, x0.1 และ x0.01 มีความคลาดเคลื่อนเท่ากับ 5.1%, 3.9% และ 4.7% ตามลำดับ

ภาควิชาวิศวกรรมนิวเคลียร์..... ลายมือชื่อนิสิต.....
 สาขาวิชานิวเคลียร์เทคโนโลยี..... ลายมือชื่อ อ.ที่ปริกษาวิทยานิพนธ์หลัก.....
 ปีการศึกษา 2554..... ลายมือชื่อ อ.ที่ปริกษาวิทยานิพนธ์ร่วม

5170382721 : MAJOR NUCLEAR ENGINEERING

KEYWORDS : GAMES PLAYER / CsI(Tl) / GAMMA SPECTROMETER / SCA

PIYATHEP CHOBTHUMKIT: DEVELOPMENT OF A GAMMA SPECTROMETER USING A PORTABLE GAMES PLAYER. ADVISOR: DECHO THONG-ARAM, CO-ADVISOR: ASST. PROF. SUVIT PUNNACHAIYA, 100 pp.

This thesis was aimed to develop a gamma spectrometer operating on a portable game player. The system was designed to be capable of both gamma spectrometry mode with energy spectrum display and dose rate mode, like the radiation survey meter, which could displayed a count rate in cps unit and an exposure rate in the range of 0-100 mR/h with 3 divided sub-ranges: x1, x0.1 and x0.01. The manipulation of radiation measurement data relied on the cooperation between an 8 bit microcontroller and a PSD integrated circuit to interface with a portable games player. The radiation detection system was designed to match with a 1cm x 1 cm CsI(Tl) scintillation crystal coupled to PIN photodiode. It has 5 main circuits: low voltage power supply, pulse amplifier, single channel analyzer (SCA), pulse counter with programmable timer, and interfacing circuit for transferring data and control signal between the portable games player and PSD. The pulse counter with programmable timer and SCA operation were controlled by a microcontroller.

The portable games player was very helpful for reducing the hardware requirement for display and giving flexibility on the graphic design. The spectroscopy mode had 12.9% energy resolution at 662 keV, tested with Cs-137 gamma source, and the dose rate mode was capable of the maximum count rate of 10 kcps. The exposure rate errors were found to be 5.1%, 3.9% and 4.7% in the sub-range of .x1, x0.1 and x0.01, respectively.

Department : Nuclear Engineering	Student's Signature
Field of Study : Nuclear Technology	Advisor's Signature
Academic Year : 2011	Co-advisor's Signature

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ผู้เขียนขอกราบขอบพระคุณ อาจารย์ เดโช ทองอร่าม ผู้ช่วยศาสตราจารย์สุวิทย์ ปุณณชัยยะ ที่ได้ประสิทธิประสาทวิชาและองค์ความรู้ ต่างๆ ให้คำปรึกษาชี้แนะ อีกทั้งให้โอกาสในการศึกษาและทำวิทยานิพนธ์นี้

ขอขอบคุณบัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย ที่ให้การสนับสนุนเงิน ทุนอุดหนุนวิทยานิพนธ์สำหรับนิสิตในการทำวิจัยนี้

ขอขอบพระคุณคณาจารย์ภาควิชาวิศวกรรมนิวเคลียร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัยทุกท่านที่ให้โอกาสในการศึกษาในครั้งนี้

ขอขอบคุณศูนย์เชี่ยวชาญนิวเคลียร์เทคโนโลยีสำหรับวิเคราะห์และทดสอบวัสดุ ภาควิชาวิศวกรรมนิวเคลียร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ที่อำนวยความสะดวก สะดวกด้านเครื่องมือและอุปกรณ์ต่างๆ สำหรับการดำเนินงานวิจัยตลอดมา

ขอขอบคุณ คุณเฉลิมพงษ์ โพธิ์ดี เจ้าหน้าที่ความปลอดภัยทางด้านรังสีระดับสูง ที่ช่วยเหลือในการปรับเทียบเครื่องมือวัด

ท้ายสุดนี้ ผู้เขียนขอกราบขอบพระคุณ คุณพ่อ คุณแม่ ที่ให้ความรักให้การอบรม สั่งสอน ให้คำปรึกษา และเป็นกำลังใจให้ข้าพเจ้าเสมอมา ขอขอบคุณพี่ชาย และคุณพิมพ์ฉัตร ราโชภาณจน์ ที่เป็นกำลังใจให้กับข้าพเจ้าเสมอมา

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ง
บทคัดย่อภาษาอังกฤษ	จ
กิตติกรรมประกาศ	ฉ
สารบัญ	ช
สารบัญตาราง	ฎ
สารบัญภาพ	ฐ
บทที่	
1. บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของการวิจัย	2
1.3 ขอบเขตของการวิจัย	2
1.4 ขั้นตอนและวิธีดำเนินงานวิจัย	2
1.5 ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย	2
1.6 เอกสารและงานวิจัยที่เกี่ยวข้อง	3
2. รังสีและระบบวัดรังสี	4
2.1 ความรู้พื้นฐานเกี่ยวกับรังสี	4
2.1.1 รังสีที่ก่อให้เกิดไอออน	4
2.1.2 รังสีที่ไม่ก่อให้เกิดไอออน	4
2.2 หน่วยวัดปริมาณรังสี	4
2.2.1 ปริมาณกัมมันตภาพรังสี	4
2.2.2 ปริมาณรังสีที่ถูกดูดกลืน (Absorbed dose)	5
2.2.3 ปริมาณรังสีที่ทำให้อากาศแตกตัว (Exposure)	5
2.2.4 ปริมาณรังสีสมมูลย์ (dose equivalent)	6
2.3 หลักการป้องกันอันตรายจากรังสีเบื้องต้น	7
2.4 ระบบวัดรังสี	8
2.4.1 ส่วนประกอบที่สำคัญของระบบวัดรังสี	9

บทที่	หน้า
2.4.2 การจัดระบบวัตรังสี	10
2.4.2.1 ระบบวัตรังสีแบบนับรวม	10
2.4.2.2 ระบบวัตรังสีแบบนับแยก	10
2.4.2.3 ระบบวัตรังสีแบบโคอินซิเดนซ์	10
2.5 เครื่องวัดสเปกตรัมรังสีแกมมา	11
2.5.1 หัววัตรังสีชนิดซิลิคิลเลชันที่ทำงานร่วมกับพินโฟโตไดโอด	11
2.5.2 วงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดียว	12
2.6 ส่วนเชื่อมโยงสัญญาณ	16
2.6.1 โปรแกรมเมเบิลซิสเต็มดีไวซ์ (พีเอสดี)	16
2.6.1.1 หน่วยความจำ	17
2.6.1.2 โปรแกรมเมเบิลลอจิกดีไวซ์	17
2.6.1.3 อินพุตและเอาต์พุตพอร์ต	18
2.6.1.4 การโปรแกรมข้อมูลผ่านทางพอร์ต JTAG	18
2.6.1.5 ส่วนเชื่อมโยงสัญญาณกับไมโครโพรเซสเซอร์หรือไมโครคอนโทรลเลอร์ ..	18
2.6.2 โปรแกรมพีเอสดีซอฟต์แวร์เพอร์ซ PSDSoft express program	20
2.6.3 ไมโครคอนโทรลเลอร์	21
2.7 เครื่องเล่นเกมบอย	23
3. วิธีดำเนินการวิจัย	27
3.1 ข้อมูลพื้นฐานในการออกแบบเครื่องเกมมาสเปกโตรมิเตอร์	27
3.1.1 ข้อมูลพื้นฐานสำหรับหัววัตรังสีและวงจรรายสัญญาณพัลส์ส่วนหน้า	27
3.1.2 ข้อมูลพื้นฐานสำหรับการออกแบบระบบวัตรังสี	28
3.1.3 ข้อมูลพื้นฐานสำหรับระบบควบคุมและแสดงผล	28
3.2 การออกแบบและพัฒนาเครื่องเกมมาสเปกโตรมิเตอร์	30
3.2.1 วงจรรายสัญญาณพัลส์	31
3.2.2 วงจรแหล่งจ่ายไฟฟ้าคิกดาต้า	32
3.2.3 วงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดียว	32
3.3 การพัฒนาส่วนเชื่อมโยงสัญญาณและส่วนแสดงผล	34

บทที่	หน้า
3.3.1 วงจรนับจำนวนพัลส์พร้อมโปรแกรมตั้งเวลา.....	34
3.3.2 โปรแกรมประมวลผลและแสดงผลบนหน้าจอของเครื่องเล่นเกมบอย	35
4. การทดสอบสมรรถนะของระบบ	39
4.1 การทดสอบวงจรแหล่งจ่ายไฟฟ้าสัปดาห์ต่ำ	39
4.1.1 เครื่องมือและอุปกรณ์	39
4.1.2 ขั้นตอนการทดสอบ	40
4.1.3 ผลการทดสอบ	40
4.2 การทดสอบวงจรขยายสัญญาณพัลส์ส่วนหน้า	41
4.2.1 เครื่องมือและอุปกรณ์	41
4.2.2 ขั้นตอนการทดสอบ	42
4.2.3 ผลการทดสอบ	42
4.3 การทดสอบวงจรขยายสัญญาณพัลส์	43
4.3.1 เครื่องมือและอุปกรณ์	43
4.3.2 ขั้นตอนการทดสอบ	43
4.3.3 ผลการทดสอบ	44
4.4 การทดสอบวงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดี่ยว	46
4.4.1 เครื่องมือและอุปกรณ์	46
4.4.2 ขั้นตอนการทดสอบ	47
4.4.3 ผลการทดสอบ	47
4.5 การทดสอบโปรแกรมตั้งเวลาและนับจำนวนสัญญาณพัลส์	48
4.5.1 เครื่องมือและอุปกรณ์	48
4.5.2 ขั้นตอนการทดสอบ	49
4.5.3 ผลการทดสอบ	49
4.6 การทดสอบความเป็นเชิงเส้นในการแปลงสัญญาณพัลส์	50
4.6.1 เครื่องมือและอุปกรณ์	50
4.6.2 ขั้นตอนการทดสอบ	51
4.6.3 ผลการทดสอบ	51

บทที่	หน้า
4.7 การทดสอบเครื่องแกมมาสเปกโตรมิเตอร์ โหมดสำรวจรังสี	52
4.7.1 เครื่องมือและอุปกรณ์	52
4.7.2 ขั้นตอนการทดสอบ	53
4.7.3 ผลการทดสอบ	54
4.8 การทดสอบเครื่องแกมมาสเปกโตรมิเตอร์ โหมดสแกน	54
4.8.1 เครื่องมือและอุปกรณ์	54
4.8.2 ขั้นตอนการทดสอบ	55
4.8.3 ผลการทดสอบ	56
4.9 การทดสอบเครื่องแกมมาสเปกโตรมิเตอร์ โหมดสวิตช์	57
4.9.1 เครื่องมือและอุปกรณ์	57
4.9.2 ขั้นตอนการทดสอบ	57
4.9.3 ผลการทดสอบ	58
5. สรุปผลการวิจัย และข้อเสนอแนะ	60
5.1 สรุปผลการวิจัย	60
5.1.1 ผลการทดสอบแหล่งจ่ายไฟฟ้าคัทดาวน์	60
5.1.2 ผลการทดสอบวงจรมายสัญญาณพัลส์ส่วนหน้า	60
5.1.3 ผลการทดสอบวงจรมายสัญญาณพัลส์	60
5.1.4 ผลการทดสอบวงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์ แบบช่องเดียว	60
5.1.5 ผลการทดสอบโปรแกรมตั้งเวลาและนับจำนวนสัญญาณพัลส์	61
5.1.6 ผลการทดสอบความเป็นเชิงเส้นในการแปลงสัญญาณพัลส์	61
5.1.7 ผลการทดสอบเครื่องแกมมาสเปกโตรมิเตอร์ โหมดสำรวจรังสี	61
5.1.8 ผลการทดสอบเครื่องแกมมาสเปกโตรมิเตอร์ โหมดสแกน	61
5.1.9 ผลการทดสอบเครื่องแกมมาสเปกโตรมิเตอร์ โหมดสวิตช์	61
5.2 ปัญหาและอุปสรรค	62
5.3 ข้อเสนอแนะ	62
รายการอ้างอิง	63

บทที่	หน้า
ภาคผนวก	64
ภาคผนวก ก คู่มือการใช้งานเครื่องแกมมาสเปกโตรมิเตอร์ ที่ทำงานร่วมกับเครื่องเล่นเกมบอย	65
ภาคผนวก ข โปรแกรมควบคุมและแสดงผลของเครื่องแกมมาสเปกโตรมิเตอร์	73
ภาคผนวก ค วงจรขยายสัญญาณพัลส์ส่วนหน้า HAMAMATSU (H4083)	97
ประวัติผู้เขียนวิทยานิพนธ์	100

สารบัญตาราง

	หน้า
ตารางที่ 2.1 ค่า Radiation weighting factor (WR)	7
ตารางที่ 2.2 หน่วยความจำภายในไอซีพีเอสดีและสัญญาณเลือกหน่วยความจำ จากตัวถอดรหัสพีแอลดี	17
ตารางที่ 2.3 รายละเอียดของเครื่องเล่นเกมบอยรุ่นต่างๆ	24
ตารางที่ 2.4 รายละเอียดของเครื่องเล่นเกมบอยรุ่นเกมบอยแอดวานซ์เอสพี	25
ตารางที่ 3.1 ฟังก์ชันการทำงานของวงจรมอนิเตอร์อินซีแดนซ์	33
ตารางที่ 4.1 อัตราการใช้กำลังไฟฟ้าสูงสุดของแต่ละแหล่งจ่ายไฟฟ้าคิกดาต้า	40
ตารางที่ 4.2 ขนาดความสูงของสัญญาณพัลส์ที่ได้จากวงจรมายสัญญาณพัลส์	45
ตารางที่ 4.3 จำนวนสัญญาณพัลส์ที่นับได้	50
ตารางที่ 4.4 ช่องพลังงานที่ได้จากการแปลงสัญญาณพัลส์ของระบบ	52
ตารางที่ 4.5 ปริมาณรังสีที่นับวัดได้ ณ จุดปรับเทียบต่างๆ ของเครื่องสำรวจรังสีที่พัฒนาขึ้น	54

สารบัญภาพ

		หน้า
ภาพที่ 2.1	แผนภาพของกระบวนการวัดรังสี	9
ภาพที่ 2.2	สัญลักษณ์และโครงสร้างของซิลิกอนฟิโนไฟโตไดโอด	12
ภาพที่ 2.3	ย่านความยาวคลื่นของแสงที่ถูกปลดปล่อยออกมาจากผลึก CsI(Tl)	12
ภาพที่ 2.4	แผนภาพการเกิดสัญญาณลจิกพัลส์ของอินทิกรัลดิสครีมิเนเตอร์	13
ภาพที่ 2.5	แผนภาพการเกิดสัญญาณลจิกพัลส์ที่เอาต์พุตของ SCA	13
ภาพที่ 2.6	แผนภาพการเกิดสัญญาณลจิกพัลส์แบบนอนไทม์มิง SCA	14
ภาพที่ 2.7	แผนภาพการเกิดสัญญาณลจิกพัลส์แบบไทม์มิง SCA (Peak sensing)	14
ภาพที่ 2.8	แผนภาพการทำงานของ SCA โหมดสแกน	16
ภาพที่ 2.9	แผนภาพโครงสร้างภายในของไอซี PSD813F2	16
ภาพที่ 2.10	แผนภาพการเชื่อมโยงแบบบัสดเรซชนิดไม่มัลติเพล็กซ์	19
ภาพที่ 2.11	แผนภาพการเชื่อมโยงแบบบัสดเรซชนิดมัลติเพล็กซ์	19
ภาพที่ 2.12	แผนภาพแสดงขั้นตอนการออกแบบฟังก์ชันการทำงาน	20
ภาพที่ 2.13	แผนภาพโครงสร้างภายในของไมโครคอนโทรลเลอร์ PIC16F877A	23
ภาพที่ 2.14	เครื่องเล่นเกมบอยรุ่นเกมบอยแอดวานซ์เอสพี	25
ภาพที่ 2.15	ข้อต่อสัญญาณของเครื่องเล่นเกมบอย	26
ภาพที่ 3.1	แผนภาพการทำงานของหัววัดรังสีและวงจรรขยายสัญญาณส่วนหน้า	28
ภาพที่ 3.2	แผนภาพการจัดสรรหน่วยความจำ (Memory Map) ของเครื่องเล่นเกมบอย	29
ภาพที่ 3.3	แผนภาพเครื่องเกมมาสปกโทรมิเตอร์ที่ทำงานร่วมกับเครื่องเล่นเกมบอย	30
ภาพที่ 3.4	แผนภาพการทำงานของวงจรรขยายสัญญาณพัลส์	31
ภาพที่ 3.5	วงจรรขยายสัญญาณพัลส์	31
ภาพที่ 3.6	วงจรแหล่งจ่ายไฟฟ้าคิกดาต้า	32
ภาพที่ 3.7	แผนภาพการทำงานของ SCA	33
ภาพที่ 3.8	วงจรรขยายสัญญาณพัลส์ และวงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์ แบบช่องเดี่ยวที่ประกอบเรียบร้อยแล้ว	34
ภาพที่ 3.9	แผนผังการทำงานของวงจรมับจำนวนพัลส์พร้อมโปรแกรมตั้งเวลา	35

ภาพที่ 3.10	แผนผังการทำงานของระบบประมวลผลและแสดงผลบนเครื่องเล่นเกมบอย	36
ภาพที่ 4.1	การจัดอุปกรณ์ทดสอบแหล่งจ่ายไฟฟ้าคัทดาวน์	40
ภาพที่ 4.2	สัญญาณรบกวนและริบเบิลของวงจรแหล่งจ่ายไฟฟ้าคัทดาวน์	41
ภาพที่ 4.3	การจัดอุปกรณ์ทดสอบระบบวัดส่วนหน้า	42
ภาพที่ 4.4	สัญญาณเอาต์พุตจากวงจรขยายสัญญาณพัลส์ส่วนหน้า	42
ภาพที่ 4.5	การจัดอุปกรณ์ทดสอบวงจรขยายสัญญาณพัลส์	43
ภาพที่ 4.6	สัญญาณพัลส์จากเครื่องกำเนิดสัญญาณพัลส์เลียนแบบสัญญาณจากหัววัดรังสี ...	44
ภาพที่ 4.7	สัญญาณเอาต์พุตของวงจรขยายสัญญาณพัลส์ที่พัฒนาขึ้น	44
ภาพที่ 4.8	กราฟความสัมพันธ์ระหว่างสัญญาณอินพุตและสัญญาณเอาต์พุต	45
ภาพที่ 4.9	การจัดอุปกรณ์ทดสอบวงจรวเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์ แบบช่องเดียว	47
ภาพที่ 4.10	สัญญาณเอาต์พุตของวงจรวเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์ แบบช่องเดียว	48
ภาพที่ 4.11	สัญญาณเอาต์พุตเมื่อสัญญาณอินพุตเป็นพัลส์คู่ที่เกิดใกล้กัน	48
ภาพที่ 4.12	การจัดอุปกรณ์ทดสอบโปรแกรมตั้งเวลาและนับจำนวนสัญญาณพัลส์ลอจิก	49
ภาพที่ 4.13	การจัดอุปกรณ์ทดสอบความเป็นเชิงเส้นในการแปลงสัญญาณพัลส์	51
ภาพที่ 4.14	กราฟแสดงความเป็นเชิงเส้นของการแปลงสัญญาณพัลส์	52
ภาพที่ 4.15	การจัดอุปกรณ์ทดสอบเครื่องแกมมาสเปกโตรมิเตอร์ใหม่ดสำรวจรังสี	53
ภาพที่ 4.16	การจัดอุปกรณ์ทดสอบเครื่องแกมมาสเปกโตรมิเตอร์ในโหมดสแกน	55
ภาพที่ 4.17	สเปกตรัมพลังงานรังสีโหมดสแกน จากต้นกำเนิดรังสีซีซีเอ็ม 137	55
ภาพที่ 4.18	สเปกตรัมพลังงานรังสีโหมดสแกน จากต้นกำเนิดรังสีโคบอลต์ 60	56
ภาพที่ 4.19	สเปกตรัมพลังงานรังสีโหมดสแกน จากต้นกำเนิดรังสีซีซีเอ็ม 137 และโคบอลต์ 60 ..	56
ภาพที่ 4.20	การจัดอุปกรณ์ทดสอบเครื่องแกมมาสเปกโตรมิเตอร์ในโหมดสวิตช์	57
ภาพที่ 4.21	สเปกตรัมพลังงานรังสีโหมดสวิตช์ จากต้นกำเนิดรังสีซีซีเอ็ม 137	58
ภาพที่ 4.22	สเปกตรัมพลังงานรังสีโหมดสวิตช์ จากต้นกำเนิดรังสีโคบอลต์ 60	58

ภาพที่ 4.23 สเปกตรัมพลังงานรังสีโหมดสวีพ จากต้นกำเนิดรังสีซีเซียม 137 และโคบอลต์ 60 ...59

ภาพที่ 4.24 กราฟเปรียบเทียบพลังงานจากการวัดสเปกตรัมด้วยหัววัด CsI(Tl)59

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันได้มีการนำรังสีแกมมามาใช้ประโยชน์ในงานด้านต่างๆ อย่างหลากหลาย เช่น ด้านอุตสาหกรรมการตรวจสอบรอยเชื่อมโลหะเพื่อหารอยร้าวหรือรอยร้าวของชิ้นงาน ด้านการแพทย์เกี่ยวกับการตรวจวินิจฉัยโรค การใช้รังสีรักษา และด้านการเกษตรเกี่ยวกับการกำจัดศัตรูพืช การปรับปรุงพันธุ์พืชเพื่อเพิ่มผลผลิต การถนอมผลผลิตทางการเกษตร เป็นต้น ซึ่งการดำเนินกิจกรรมในงานด้านต่างๆ เหล่านี้ อาจส่งผลกระทบต่อผู้ปฏิบัติงานและชุมชนโดยรอบไม่ว่าจะเป็นแหล่งน้ำ พืช สัตว์ ตลอดจนประชาชนที่อาศัยอยู่ในชุมชนนั้นๆ เนื่องจากไม่สามารถทราบได้ว่าในชุมชนที่อาศัยอยู่นั้นมีการแพร่กระจายของสารกัมมันตรังสีหรือไม่ เนื่องจากไม่มีเครื่องมือที่ใช้ในการตรวจวัดสารกัมมันตรังสี เพราะเครื่องมือวัดรังสีส่วนใหญ่เป็นเครื่องมือที่ผลิตจากต่างประเทศ มีราคาแพง จึงทำให้เป็นเรื่องยากในการจัดหาเครื่องมือวัดรังสีเหล่านี้ให้กับชุมชนต่างๆ เพื่อให้ไว้ใช้งาน จากเหตุผลดังกล่าวจึงเกิดแนวคิดที่จะพัฒนาเครื่องมือหรืออุปกรณ์วัดรังสีต้นแบบที่มีราคาถูกและง่ายต่อการใช้งาน ด้วยวัสดุอุปกรณ์ที่สามารถจัดหาได้ภายในประเทศ และจากการศึกษาค้นคว้างานวิจัยต่างๆ ที่เกี่ยวข้องพบว่ามีความเป็นไปได้ที่จะทำการพัฒนาเครื่องมือและอุปกรณ์วัดรังสีที่มีคุณสมบัติดังกล่าวข้างต้น

จากการศึกษาพบว่าเครื่องเล่นเกมส์แบบพกพา เช่นเครื่องเล่นเกมส์บอย ซึ่งเป็นเครื่องเล่นที่ได้รับความนิยมอย่างแพร่หลาย ใช้งานง่าย และสะดวกแก่การพกพา สามารถทำการเปลี่ยนตลับเกมส์ได้อย่างอิสระ ประกอบกับได้มีงานวิจัยต่างๆ ที่นำเครื่องเล่นเกมส์บอยมาทำการพัฒนาร่วมกับระบบวัดสัญญาณทางด้านอิเล็กทรอนิกส์ที่อยู่ในรูปแบบตลับเกมส์ ซึ่งแสดงให้เห็นว่าเครื่องเล่นเกมส์บอยมีความสามารถในการประมวลผลและมีความละเอียดของหน้าจอเพียงพอต่อการแสดงผล จากเหตุผลดังกล่าวผู้วิจัยจึงมีแนวคิดในการพัฒนาเครื่องเกมส์พกพาโทรมิเตอร์ที่อยู่ในรูปแบบของตลับเกมส์สำหรับเชื่อมต่อกับเครื่องเล่นเกมส์บอยที่สามารถนับอัตรารังสีเฉลี่ยและแสดงสเปกตรัมรังสีแกมมาได้

1.2 วัตถุประสงค์ของการวิจัย

เพื่อพัฒนาเครื่องวัดสเปกตรัมรังสีแกมมาโดยใช้เครื่องเล่นเกมส์แบบพกพา

1.3 ขอบเขตของการวิจัย

1. ออกแบบและสร้างส่วนประกอบต่างๆ ของเครื่องวัดสเปกตรัมรังสีแกมมา ประกอบด้วย วงจรแหล่งจ่ายไฟฟ้าคักคาต่ำ วงจรขยายสัญญาณพัลส์ วงจรวิเคราะห์ ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดียว และส่วนเชื่อมโยงสัญญาณกับเครื่องเล่นเกมส์
2. พัฒนาโปรแกรมที่ใช้ในการควบคุมและแสดงผลสเปกตรัมรังสีแกมมาและอัตราปริมาณรังสีที่วัดได้
3. ทดสอบสมรรถนะการทำงานของระบบ ได้แก่ ทดสอบอัตรานับรังสีเฉลี่ยสูงสุดของระบบ และการวัดสเปกตรัมรังสีแกมมา

1.4 ขั้นตอนและวิธีดำเนินงานวิจัย

1. ศึกษาค้นคว้าเอกสารและงานวิจัยที่เกี่ยวข้อง
2. ออกแบบและสร้างส่วนประกอบต่างๆ ของเครื่องวัดสเปกตรัมรังสีแกมมา ประกอบด้วย วงจรแหล่งจ่ายไฟฟ้าคักคาต่ำ วงจรขยายสัญญาณพัลส์ วงจรวิเคราะห์ ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดียว และส่วนเชื่อมโยงสัญญาณกับเครื่องเล่นเกมส์
3. พัฒนาโปรแกรมที่ใช้ในการควบคุมและแสดงผลสเปกตรัมรังสีแกมมาและอัตราปริมาณรังสีที่วัดได้
4. ทดสอบสมรรถนะการทำงานของระบบ ได้แก่ ทดสอบอัตรานับรังสีเฉลี่ยสูงสุดของระบบ และการวัดสเปกตรัมรังสีแกมมา
5. สรุปผลการทดลองและเขียนวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย

ได้เครื่องวัดสเปกตรัมรังสีแกมมาโดยใช้เครื่องเล่นเกมส์แบบพกพา เช่นเครื่องเล่นเกมส์ที่สามารถนับอัตรารังสีเฉลี่ย และแสดงผลสเปกตรัมรังสีแกมมา

1.6 เอกสารและงานวิจัยที่เกี่ยวข้อง

1. ปี พ.ศ. 2547 สิริยาภรณ์ แสงอรุณ [1] ทำการวิจัยเรื่องการพัฒนาระบบมอดูเลเตอร์รังสีแกมมาชนิดเคลื่อนย้ายได้ที่เชื่อมโยงด้วยจีพีเอส โดยใช้หัววัดรังสีชนิด CsI(Tl) คับปลั๊กพินโฟโตไดโอด ขนาด 10 X 10 มิลลิเมตร และขยายสัญญาณพัลส์เพื่อนำสัญญาณที่ได้ป้อนให้กับวงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดียวขนาด 256 ช่องพลังงาน และทำการแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัลขนาด 10 บิต เพื่อส่งต่อไปยังคอมพิวเตอร์ขนาดพกพาชนิดปาล์มรุ่น m505 ซึ่งระบบที่พัฒนามีความสามารถในการแจกแจงพลังงานเท่ากับ 14.28% ที่พลังงาน 611.66 keV

2. ปี พ.ศ. 2543 Steve Willis [2] เรื่อง Gameboy Digital Sampling Oscilloscope (GBDSO) เป็นการพัฒนาเครื่องเล่นเกมน้อยเพื่อใช้งานร่วมกับเครื่องแสดงคลื่นกระแสไฟฟ้า (Oscilloscope) โดยการนำเครื่องเล่นเกมน้อยรุ่นเกมบอยพ็อกเก็ต หรือเกมบอยคัลเลอร์ ออกซิไลด์โคปที่พัฒนาออกแบบให้อยู่ในรูปของตลับเกมบอยซึ่งสามารถรับสัญญาณอินพุตได้ 2 ช่องสัญญาณ คือ ช่องสัญญาณเอ (CHA) และช่องสัญญาณบี (CHB) โดยแต่ละช่องสัญญาณนั้นมีอินพุตอิมพีแดนซ์ขนาด 1 เมกกะโอห์ม สามารถเลือกอัตราการลดทอนสัญญาณได้แบบ 1:1 หรือ 10:1 สามารถเลือกโหมดการทำงานได้ 2 โหมด คือโหมดไฟฟ้ากระแสสลับ และโหมดไฟฟ้ากระแสตรง สามารถรับแรงดันอินพุตตั้งแต่ 50 มิลลิโวลต์ ถึง 10 โวลต์ ต่อหนึ่งช่องสัญญาณและอยู่ในช่วงความถี่ 0-100 กิโลเฮิรตซ์

3. ปี พ.ศ. 2549 Marcel Cremmel [3] เรื่อง Gameboy Electrocardiograph (GBECG) เป็นการพัฒนาต่อจาก GBDSO แต่เลือกใช้เครื่องเล่นเกมน้อยรุ่นเกมบอยแอดวานซ์ โดยพัฒนาให้อยู่ในรูปตลับเกมที่มีขนาดเท่ากับตลับเกมบอย โดยการนำไอซีพีเอสดี (PSD813F2) มาใช้ในส่วนของเชื่อมโยงสัญญาณระหว่างเครื่องอ่านคลื่นหัวใจกับเครื่องเล่นเกมน้อย ซึ่งสามารถปรับกำลังขยายของวงจรขยายสัญญาณได้ถึง 1000 เท่า และแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัลขนาด 8 บิต โดยทำงานที่ความถี่ 477.84 เฮิรตซ์ เพื่อนำไปพล็อตเป็นสเปกตรัมของระบบอีซีจี (ECG) สัญญาณดิจิทัลที่รับเข้ามาถูกควบคุมโดยตัวประมวลผล ซึ่งจะนำมาเก็บในหน่วยความจำขนาด 8 กิโลไบต์ และทำการอ่านซ้ำ เพื่อแสดงเป็นสเปกตรัมต่อหน่วยเวลา บนหน้าจอแสดงผลของเครื่องเล่นเกมน้อย

บทที่ 2

รังสีและระบบวัดรังสี

2.1 ความรู้พื้นฐานเกี่ยวกับรังสี [4]

รังสี (Radiation) เป็นพลังงานรูปแบบหนึ่งที่สามารถแพร่กระจายออกจากต้นกำเนิดในรูปแบบต่างๆ เช่นในรูปแบบของคลื่นแม่เหล็กไฟฟ้า ได้แก่ รังสีเอกซ์ (X-ray) รังสีแกมมา (Gamma rays) และในรูปของอนุภาค ได้แก่ รังสีแอลฟา (Alpha rays) รังสีบีตา (Beta rays) และรังสีนิวตรอน (Neutron) เป็นต้น โดยที่เราสามารถจำแนกประเภทของรังสีเป็น 2 ประเภทใหญ่ๆ คือ

2.1.1 รังสีที่ก่อให้เกิดไอออน (ionizing radiation)

รังสีที่ก่อให้เกิดไอออน คือรังสีที่ทำให้เกิดการแตกตัวเป็นประจุในตัวกลางที่รังสีเคลื่อนที่ผ่าน ได้แก่ รังสีเอกซ์ (γ) รังสีแอลฟา (α) รังสีบีตา (β) รังสีแกมมา (γ) และรังสีนิวตรอน (n) เป็นต้น

2.1.2 รังสีที่ไม่ก่อให้เกิดไอออน (non-ionizing radiation)

รังสีที่ไม่ก่อให้เกิดไอออน คือรังสีที่ไม่ทำให้เกิดการแตกตัวเป็นประจุในตัวกลางที่รังสีเคลื่อนที่ผ่าน ได้แก่ รังสีคลื่นแม่เหล็กไฟฟ้า เช่น อัลตราไวโอเล็ต และคลื่นไมโครเวฟ เป็นต้น

2.2 หน่วยวัดปริมาณรังสี [4]

เมื่อร่างกายของมนุษย์ได้รับรังสีที่มีพลังงานสูงพอจะทำให้เซลล์ภายในร่างกายเกิดการไอออนไนซ์ เป็นผลให้อะตอมหรือโมเลกุลของเซลล์ถูกทำลายและอาจทำให้เกิดอาการผิดปกติภายในร่างกาย ดังนั้นจึงได้มีการกำหนดมาตรฐานของหน่วยวัดปริมาณรังสีสำหรับการตรวจวัดและควบคุมความปลอดภัยทางด้านรังสีและกัมมันตรังสีขึ้นดังนี้

2.2.1 ปริมาณกัมมันตภาพรังสี

ปริมาณกัมมันตภาพรังสี เป็นหน่วยที่ใช้วัดปริมาณกัมมันตภาพรังสีของไอโซโทปแบบชั่วคราว ทั้งนี้เพราะว่าเมื่อเวลาผ่านไปปริมาณไอโซโทปรังสีในสารกัมมันตรังสีจะมีปริมาณลดลงพร้อมกับเกิดธาตุใหม่ขึ้น โดยเรียกปรากฏการณ์นี้ว่า การสลายตัว ซึ่งการวัดปริมาณกัมมันตภาพรังสีในลักษณะนี้จึงเป็นการวัดปริมาณรังสีที่เกิดขึ้นจากการสลายตัวของธาตุในขณะนั้น โดยหน่วยวัดปริมาณกัมมันตภาพรังสีเดิมอาศัยการอ้างอิงจากอัตราการเปลี่ยนแปลงทางนิวเคลียร์ของธาตุเรเดียมน้ำหนัก 1 กรัม ซึ่งสลายตัว 3.7×10^{10} ครั้งต่อวินาที (Disintegration per second, dps) เท่ากับ 1 คูรี (Curie, Ci)

$$1 \text{ Ci} = 3.7 \times 10^{10} \text{ dps} \quad \text{..... (2.1)}$$

ต่อมาใช้เป็น SI Unit โดยกำหนดชื่อเฉพาะของหน่วยวัดกัมมันตภาพรังสีว่า เบคเคอเรล (Bq)

โดยที่

$$1 \text{ Bq} = 1 \text{ dps} \quad \text{..... (2.2)}$$

และ

$$1 \text{ Ci} = 3.7 \times 10^{10} \text{ Bq} \quad \text{..... (2.3)}$$

2.2.2 ปริมาณรังสีที่ถูกดูดกลืน (Absorbed dose)

เนื่องจากรังสีแต่ละชนิดมีความสามารถในการทะลุผ่านวัตถุได้ต่างกัน และถ่ายเทพลังงานให้กับวัตถุแต่ละชนิดไม่เท่ากัน ดังนั้นผลของรังสีต่อวัตถุจึงแปรผันตามปริมาณพลังงานของรังสีที่วัตถุนั้นดูดกลืน ตัวอย่างเช่น รังสีแอลฟา และรังสีบีตาซึ่งเป็นรังสีที่มีพิสัยต่ำจะถ่ายเทพลังงานทั้งหมดให้กับวัตถุ เช่นเดียวกับฟิสิกส์พลังงานต่ำๆ โดยจะส่งผลให้เกิดรอยไหม้ที่ผิวหนังและถ้าเป็นฟิสิกส์พลังงานสูงหรือนิวตรอน พลังงานส่วนมากจะทะลุผ่านออกไปจากวัตถุ และจะมีเพียงบางส่วนของพลังงานเท่านั้นที่ถูกดูดกลืนไว้ โดยหน่วยของปริมาณรังสีที่ถูกดูดกลืนเดิมนั้นใช้ Radiation absorbed dose (rad) ซึ่งเท่ากับพลังงานรังสีที่ถูกดูดกลืน 100 ergs ในวัตถุมวล 1 กรัม

$$1 \text{ rad} = 100 \text{ ergs} / \text{g} \quad \text{..... (2.4)}$$

ในปัจจุบัน SI Unit ใช้หน่วยของพลังงานเป็นจูล (joule) และหน่วยของมวลเป็นกิโลกรัม (kilogram) จึงได้มีการใช้หน่วยของปริมาณรังสีที่ถูกดูดกลืนเป็นจูลต่อกิโลกรัมหรือเรียกว่า เกรย์ (Gray, Gy) โดยที่

$$1 \text{ Gy} = 1 \text{ J} / \text{kg} = 100 \text{ rad} \quad \text{..... (2.5)}$$

2.2.3 ปริมาณรังสีที่ทำให้อากาศแตกตัว (Exposure)

ปริมาณรังสีที่ทำให้อากาศแตกตัว เป็นปริมาณรังสีที่ไม่เกี่ยวข้องโดยตรงกับผลของรังสี เพราะเพียงแต่วัดว่ามีการแตกตัวของอากาศมากน้อยเพียงใดเท่านั้น ซึ่งการวัดรังสีในรูปแบบนี้นิยมใช้กันมากเนื่องจากเป็นวิธีที่มีความไวสูง และสามารถวัดค่าได้ถูกต้องแม่นยำ โดยหน่วยเดิมของปริมาณรังสีที่ทำให้อากาศแตกตัวคือ เรินแกนท์ (Roentgen, R) ซึ่งมีค่าเท่ากับพลังงานของรังสีที่ทำให้อากาศแตกตัวให้ประจุ 1 e.s.u. (electrostatic unit) ในอากาศแห้ง 1 cm³ ที่ STP หรือ

อากาศมวล 1.293×10^3 กรัม สำหรับในหน่วย SI Unit ใช้เป็นคูอมบ์ต่อกิโลกรัม (C/kg) โดย
ประจุ 1 e.s.u. มีค่าเท่ากับ 3.335×10^{-10} C

$$1R = 2.58 \times 10^{-4} \text{ C / kg} \quad \text{..... (2.6)}$$

2.2.4 ปริมาณรังสีสมมูลย์ (Dose equivalent)

ปริมาณรังสีสมมูลย์ เป็นหน่วยที่นำเอาผลทางชีววิทยาของรังสีเข้ามาเกี่ยวข้อง เนื่องจาก
รังสีแต่ละชนิดมีคุณสมบัติต่างกัน ผลกระทบต่อระบบชีววิทยาเมื่อได้รับรังสีที่ปริมาณรังสีที่ถูก
ดูดกลืนเดียวกันจึงแตกต่างกัน ปริมาณรังสีสมมูลย์นี้มีค่าเท่ากับผลคูณระหว่างปริมาณรังสีที่ถูก
ดูดกลืน (D) และค่าแฟกเตอร์คุณภาพของรังสี (quality factor, Q)

$$H_T = DQ \quad \text{..... (2.7)}$$

ปริมาณรังสีสมมูลย์มีหน่วยเป็น เรม (Roentgen equivalent man, rem) โดยที่ปริมาณ
รังสี 1 เรม หมายถึงปริมาณรังสีที่ก่อให้เกิดผลทางชีววิทยาเทียบเท่ากับผลที่เกิดจากรังสีเอกซ์หรือ
รังสีแกมมา 1 แรด ต่อมาใช้เป็น SI Unit ได้มีการเปลี่ยนหน่วยปริมาณรังสีสมมูลย์เป็น ซีเวิร์ต
(Sievert, Sv) โดยที่

$$1\text{Sv} = 100 \text{ rem} \quad \text{..... (2.8)}$$

ในปัจจุบัน ICRP ได้เปลี่ยนค่าแฟกเตอร์คุณภาพของรังสีเป็น Radiation weighting
factor (W_R) แทน และใช้ equivalent dose แทน dose equivalent ดังนั้นในการหาผลรวมของ
equivalent dose (H_T) ของกลุ่มเนื้อเยื่อ และอวัยวะต่างๆ สามารถหาได้จากสมการดังต่อไปนี้

$$H_T = \sum_R W_R D_{TxR} \quad \text{..... (2.9)}$$

เมื่อ D_{TxR} คือรังสีที่ถูกดูดกลืนเฉลี่ยทั่วบริเวณกลุ่มเนื้อเยื่อหรืออวัยวะ (T) เนื่องจากรังสี
(R) และค่า W_R จะมีความสัมพันธ์กับค่า Relative biological effectiveness (RBE) โดยอาศัย
การเปรียบเทียบความเสียหายของเนื้อเยื่อ เมื่อเนื้อเยื่อถูกดูดกลืนรังสีที่ต่างชนิด และต่างพลังงาน
แสดงในตารางที่ 2.1

ตารางที่ 2.1 ค่า Radiation weighting factor (W_R)

Type and energy range	Radiation weighting factor (W_R)
Photon, all energy	1
Electrons and muons, all energy	1
Neutrons, energy < 10 keV	5
> 10 keV to 100 keV	10
> 100 keV to 2 MeV	20
> 2 MeV to 20 MeV	10
> 20 MeV	5
Protons, other than recoil proton, energy > 2 MeV	5
Alpha particles, fission fragments, heavy nuclei	20

2.3 หลักการป้องกันอันตรายจากรังสีเบื้องต้น [4]

ในการปฏิบัติงานเกี่ยวกับรังสีนั้น ความปลอดภัยเป็นสิ่งสำคัญที่จำเป็นต้องคำนึงถึงอยู่เสมอไม่ว่าผู้ปฏิบัติงานจะทำงานเกี่ยวข้องกับรังสีชนิดใดก็ตาม การประเมินความอันตรายจากรังสีในขณะที่ทำงานมีหลักเกณฑ์ง่าย ๆ คือ “ในการปฏิบัติงานทางรังสีใดๆ ผู้ปฏิบัติงานจะต้องได้รับรังสีไม่เกินระดับปริมาณรังสีที่ยอมรับได้ (Dose limits)” ซึ่งหมายถึงระดับของปริมาณรังสีที่ยอมให้ผู้ปฏิบัติงานรับได้สูงสุดต่อช่วงระยะเวลาที่ปฏิบัติงาน ซึ่งถูกกำหนดขึ้นโดยคณะกรรมการระหว่างประเทศว่าด้วยการป้องกันอันตรายจากรังสี หรือ ICRP (International Commission of Radiological Protection) ซึ่งคณะกรรมการนี้จะดูแลปรับปรุงข้อกำหนดเกี่ยวกับระดับรังสีที่ยอมรับให้รับได้อยู่เสมอตามความเจริญก้าวหน้าทางวิชาการ กลุ่มแรกคือผู้ปฏิบัติงานเกี่ยวข้องกับรังสี อีกกลุ่มคือประชาชนทั่วไป ในทางปฏิบัตินั้นผู้ปฏิบัติงานทางรังสีจะต้องยึดหลักในการป้องกันอันตรายจากรังสีเพื่อควบคุมให้ตนเองและผู้อื่นได้รับปริมาณรังสีน้อยที่สุดเท่าที่จะเป็นไปได้

การป้องกันอันตรายจากรังสีนั้นมีปัจจัยที่จะต้องยึดถือเป็นหลักปฏิบัติด้วยกัน 3 ประการ คือ ระยะเวลาที่ได้รับรังสี ระยะห่างระหว่างผู้ปฏิบัติงานกับต้นกำเนิดรังสี และการใช้วัสดุกำบังรังสี

1. เวลา (Time) ปริมาณรังสีที่ผู้ปฏิบัติงานได้รับนั้นจะขึ้นอยู่กับเวลาและอัตราปริมาณรังสีที่ได้รับ ซึ่งมีความสัมพันธ์ดังแสดงในสมการที่ 2.10

$$\text{total dose} = \text{dose rate} \times \text{exposure time} \dots\dots\dots (2.10)$$

เมื่อ total dose = ปริมาณรังสีที่ได้รับทั้งหมด
 dose rate = อัตราปริมาณรังสีในบริเวณที่ปฏิบัติงาน
 exposure time = เวลาที่ได้รับรังสี

จากความสัมพันธ์จะพบว่า ถ้าใช้เวลาในการปฏิบัติงานทางรังสีให้น้อยลงจะช่วยให้ได้รับรังสีในปริมาณที่น้อยลงด้วยเช่นกัน

2. ระยะทาง (Distance) ปริมาณความเข้มของรังสี ณ บริเวณใดๆ เป็นสัดส่วนผกผันกำลังสองกับระยะห่างจากต้นกำเนิดรังสี โดยมีลักษณะความสัมพันธ์ดังนี้

$$I \propto \frac{1}{d^2} \dots\dots\dots (2.11)$$

เมื่อ I = ปริมาณรังสี ณ บริเวณใดๆ
 d = ระยะห่างจากต้นกำเนิดรังสี

จากความสัมพันธ์พบว่าปริมาณรังสีจะลดลงเมื่อปฏิบัติงานห่างจากต้นกำเนิดรังสีมากขึ้น

3. วัสดุกำบังรังสี (Shielding) เมื่อรังสีผ่านตัวกลางใดๆ จะมีปริมาณลดลงตามกฎของแลมเบิร์ต (Lambert's Law) ดังนี้

$$I = I_0 e^{-\mu x} \dots\dots\dots (2.12)$$

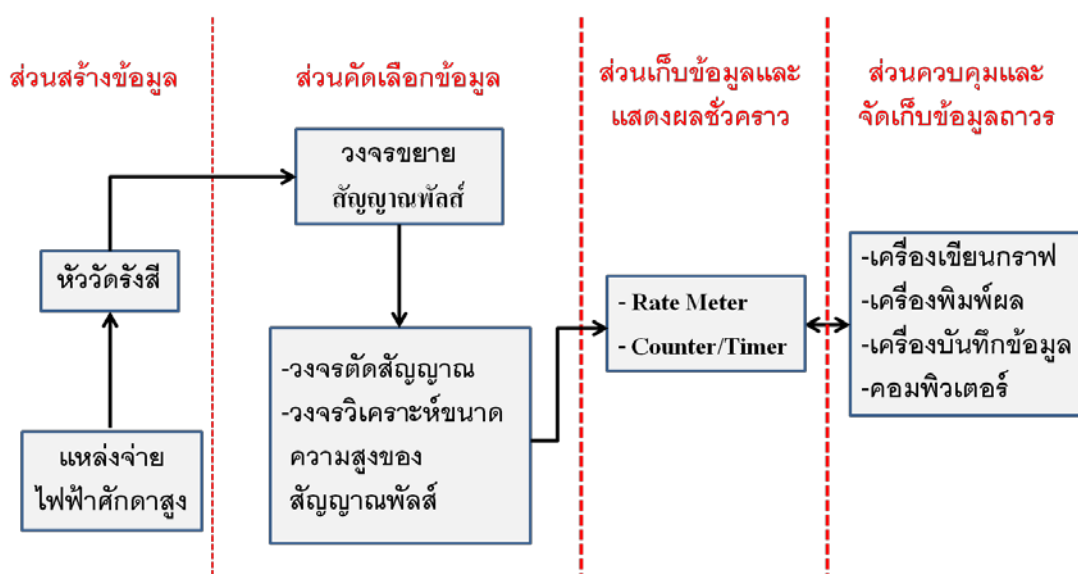
เมื่อ I = ปริมาณรังสีหลังผ่านวัสดุกำบังรังสี
 I_0 = ปริมาณรังสีก่อนผ่านวัสดุกำบังรังสี
 μ = สัมประสิทธิ์การลดทอนปริมาณรังสี
 x = ความหนาของวัสดุกำบังรังสี

ดังนั้นการปฏิบัติงานหลังวัสดุกำบังรังสีจะช่วยให้ผู้ปฏิบัติงานได้รับปริมาณรังสีน้อยลง

2.4 ระบบวัดรังสี

ระบบวัดรังสีเป็นกระบวนการจัดการปริมาณทางฟิสิกส์ของรังสีให้อยู่ในรูปแบบของปริมาณสัญญาณทางไฟฟ้าด้วยระบบจัดการสัญญาณทางอิเล็กทรอนิกส์เฉพาะหน้าที่ ซึ่งเริ่มจากการแปลงปริมาณประจุไฟฟ้าจากหัววัดรังสีให้เป็นสัญญาณอนาลอกที่แผ่ด้วยรายละเอียดของพลังงานรังสีที่

ได้รับการถ่ายโอนในหัววัดรังสี และอัตรานับรังสีในลักษณะของรูปสัญญาณพัลส์พร้อมที่จะนำไปทำการคัดเลือกขนาดของสัญญาณหรือลักษณะของรูปสัญญาณ ด้วยการทำงานทางสัญญาณเชิงเส้น (Linear Signal) หรือสัญญาณเชิงตัวเลข (Digital Signal) ตามความเหมาะสม สัญญาณที่ได้รับการคัดเลือกแล้วจะส่งเข้าสู่ขั้นตอนการนับปริมาณสัญญาณที่เกิดจากการวัดรังสีทั้งในรูปแบบของการนับที่แจกแจงพลังงาน และการนับที่ไม่แจกแจงพลังงาน ดังแสดงในภาพที่ 2.1 การวัดรังสีมีลักษณะเป็นการนับจำนวนรังสีที่ผ่านการคัดเลือกแล้วจึงเรียกระบบวัดรังสีว่า ระบบนับรังสี (Radiation Counting System)



ภาพที่ 2.1 แผนภาพของระบบการวัดรังสี

2.4.1 ส่วนประกอบที่สำคัญของระบบวัดรังสี [5]

ระบบวัดรังสีโดยทั่วไปสามารถแบ่งหน้าที่การทำงานของกลุ่มอุปกรณ์การวัดเฉพาะหน้าที่สำหรับการจัดการสัญญาณได้เป็น 4 ส่วนหลักๆ คือ

ก. ส่วนสร้างข้อมูล (Data Production) เป็นส่วนการทำงานที่ประกอบด้วยหัววัดรังสีที่ได้รับการไบอัสด้วยแหล่งจ่ายไฟฟ้าศักดาสูง (High Voltage) ทำหน้าที่ตรวจวัดรังสีและสร้างปริมาณสัญญาณไฟฟ้าทั้งแบบที่เป็นสัดส่วนโดยตรงต่อการถ่ายโอนพลังงานของรังสีในตัวกลางของหัววัดรังสี เช่น หัววัดรังสีแบบซิลทิลเลชัน และแบบที่ไม่เป็นสัดส่วนต่อการถ่ายโอนพลังงาน เช่น หัววัดรังสีแบบไกเกอร์มูลเลอร์ เป็นต้น

ข. ส่วนคัดเลือกข้อมูล (Data Selection) เป็นส่วนการทำงานที่ประกอบด้วยวงจรมายาสัญญาณพัลส์ (Pulse Amplifier) และวงจรถัดสัญญาณที่ไม่ต้องการ (Discriminator) หรือเลือก

ช่วงสัญญาณที่ต้องการวิเคราะห์ ทั้งในรูปแบบของขนาดความสูงของสัญญาณพัลส์และช่วงเวลาที่แตกต่างกันของสัญญาณลอจิกที่ได้รับการแปลงผันหลังเกิดอันตรกิริยาภายในหัววัดรังสี ข้อมูลจากส่วนการทำงานนี้จะส่งเข้ากระบวนการวิเคราะห์และเก็บข้อมูลต่อไป

ค. ส่วนเก็บข้อมูล (Data Accumulation) เป็นส่วนการทำงานที่ประกอบด้วยวงจรสำหรับทำหน้าที่สะสมข้อมูลจำนวนนับรังสีรวม (Integral counting system) ได้แก่ อุปกรณ์นับรังสี (Counter or Scaler) หรืออุปกรณ์วัดค่าอัตรานับรังสี (Rate meter) และการนับรังสีเฉพาะค่าพลังงาน (Differential counting system) ซึ่งอุปกรณ์วิเคราะห์ขนาดความสูงของสัญญาณพัลส์ (Pulse Height Analyzer; PHA) เป็นต้น โดยอุปกรณ์เหล่านี้จะทำการเก็บข้อมูลจำนวนนับรังสีเพื่อแสดงผลชั่วคราวทั้งในรูปแบบที่เป็นตัวเลขที่แสดงค่าของจำนวนนับหรืออาจมีการจัดการข้อมูลให้แสดงผลอยู่ในรูปของสเปกตรัมพลังงานหรือสเปกตรัมเวลาได้

ง. ส่วนควบคุมการทำงานของระบบและส่วนจัดเก็บข้อมูล (System Control and Data Handling) ในส่วนของอุปกรณ์ควบคุมการทำงานของระบบจะจัดการด้านการควบคุมระยะเวลาในการวัดรังสี ควบคุมการวิเคราะห์สัญญาณพัลส์ของการวัดรังสี โดยจะเป็นตัวสร้างสัญญาณควบคุมที่จำเป็นเพื่อส่งไปควบคุมการทำงานของวงจรในส่วนต่างๆ ตามความเหมาะสม สำหรับส่วนการจัดเก็บข้อมูลแบบถาวรคืออุปกรณ์บันทึกข้อมูลถาวร ได้แก่ เครื่องเขียนกราฟ เครื่องพิมพ์ผล เครื่องบันทึกข้อมูลแบบแผ่นจานแม่เหล็ก และคอมพิวเตอร์ เป็นต้น

2.4.2 การจักระบบวัดรังสี [5]

การจักระบบวัดรังสีเฉพาะงานแบ่งได้ 3 ประเภท ตามลักษณะของข้อมูลการวัดรังสี ดังนี้

2.4.2.1 ระบบวัดรังสีแบบนับรวม (Integral counting system) เป็นระบบวัดรังสีที่จัดอุปกรณ์วัด เพื่อการตรวจนับรังสีทุกพลังงานที่ตรวจวัดได้ กล่าวคือ นับรังสีรวมไม่แยกนับเฉพาะพลังงาน เช่น การวัดความแรงรังสี (Activity) การวัดความเข้มรังสี (Intensity) ในหน่วยแสดงผลจำนวนนับต่อเวลา (Count per minute: CPM หรือ Count per second: CPS)

2.4.2.2 ระบบวัดรังสีแบบนับแยก (Differential counting system) เป็นระบบวัดรังสีที่จัดอุปกรณ์วัดสำหรับวัดรังสีแบบแยกนับเฉพาะพลังงานที่สนใจ แสดงผลในรูปแบบของสเปกตรัมพลังงาน (Energy spectrum) เช่น ระบบวิเคราะห์พลังงานของรังสีแกมมา (γ -spectrometer) และวิเคราะห์รังสีอัลฟา (α -spectrometer) เป็นต้น

2.4.2.3 ระบบวัดรังสีแบบโคอินซิเดนซ์ (Co-Incidence counting system) เป็นระบบวัดรังสีที่จัดอุปกรณ์สำหรับวัดจำนวนนับรังสีแยกเฉพาะช่วงเวลา ที่มีการแสดงผลอยู่ในรูปของสเปกตรัม

เวลา (Time spectrum) เช่นระบบวัดแบบฟาสต์โคอินซิเดนซ์ (Fast co-incidence counting system) ระบบวัดแบบสโลว์โคอินซิเดนซ์ (Slow co-incidence counting system) เป็นต้น และสามารถประยุกต์ใช้กับระบบวัดรังสีที่มีรังสีรบกวนต่ำ (Low background counting system)

2.5 เครื่องวัดสเปกตรัมรังสีแกมมา

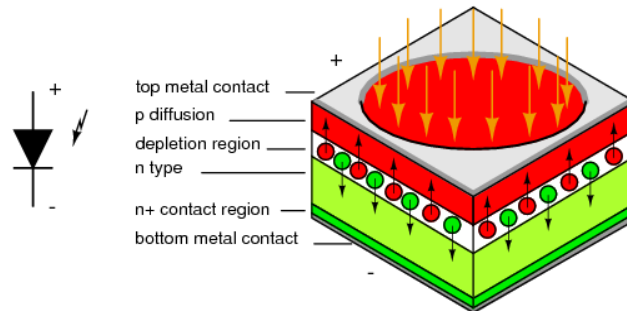
เครื่องวัดสเปกตรัมของรังสีแกมมาเป็นอุปกรณ์วัดรังสีที่มีการจัดระบบวัดรังสีแบบนับแยกพลังงาน โดยมีโครงสร้างการทำงานของระบบที่ประกอบด้วย หัววัดรังสีชนิดซินทิลเลชันหรือหัววัดรังสีชนิดอื่นที่สามารถแยกแกระดับพลังงานของรังสีที่ตกกระทบบนหัววัดได้ แหล่งจ่ายไฟฟ้าสำหรับการไบอัสหัววัดรังสี วงจรขยายสัญญาณพัลส์ วงจรวิเคราะห์ความสูงของสัญญาณพัลส์ และวงจรมันทีกข้อมูลพร้อมการแสดงผลในรูปแบบสเปกตรัมของพลังงาน

2.5.1 หัววัดรังสีชนิดซินทิลเลชันที่ทำงานร่วมกับฟิโนโตไดโอด [1]

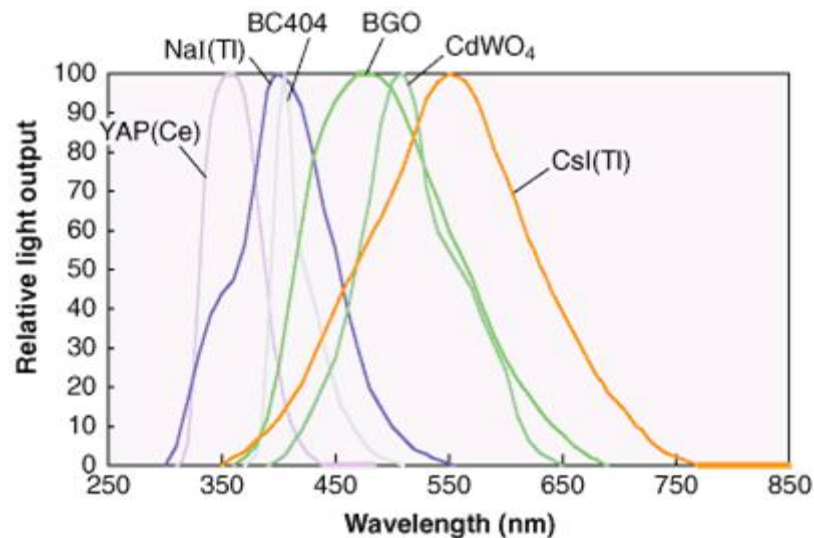
โครงสร้างของหัววัดรังสีชนิดซินทิลเลชันที่ทำงานร่วมกับฟิโนโตไดโอด (PIN Photodiode scintillation detector) คือ การนำซิลิกอนฟิโนโตไดโอด (Silicon PIN photodiode) ดังภาพที่ 2.2 มาประยุกต์ใช้ในการตรวจวัดประกายแสง (Scintillated light) ที่เกิดขึ้นบนผลึกซินทิลเลเตอร์ชนิดซีเซียมไอโอไดด์ซึ่งแตกตัวด้วยทาลเลียม (CsI(Tl)) เมื่อมีรังสีมาตกกระทบโดยทั่วไปผลึก CsI(Tl) มีคุณสมบัติในการดูดกลืนอนุภาคของ รังสีเอกซ์ และรังสีแกมมาได้ดี โดยมีการปลดปล่อยแสงที่มีความยาวคลื่นในช่วงตั้งแต่ 350 นาโนเมตร ถึง 750 นาโนเมตร แสดงดังภาพที่ 2.3 ซึ่งจำนวนครั้งของการเกิดแสงและปริมาณความเข้มของแสงที่เกิดขึ้นบนผลึก CsI(Tl) ขึ้นอยู่กับความแรงรังสีและพลังงานของรังสีที่ตกกระทบ สำหรับฟิโนโตไดโอดทำหน้าที่เป็นตัวรับแสงที่เกิดขึ้นแล้วเปลี่ยนเป็นสัญญาณไฟฟ้า ก่อนส่งไปยังวงจขยายสัญญาณส่วนหน้า ชนิดที่มทำงานแบบไวต่อประจุ (Charge Sensitive Pre-Amplifier) และวงจขยายสัญญาณพัลส์ เพื่อขยายสัญญาณพัลส์นิวเคลียร์ให้มีขนาดใหญ่ขึ้นและเหมาะสมกับความต้องการของระบบวัดรังสีที่อยู่ในส่วนถัดไป

สำหรับข้อดีของการใช้ฟิโนโตไดโอด คือ สนามแม่เหล็กจะไม่มีผลกระทบต่อการทำงานเมื่อเปรียบเทียบกับ PMT (Photo Multiplier Tube) จึงทำให้มีเสถียรภาพต่อการทำงานที่ดี และยังสร้างเครื่องมือที่มีขนาดเล็กลงได้ เพราะไม่จำเป็นต้องใช้แหล่งจ่ายไฟฟ้าศักดาสูง (High voltage power supply) แต่มีข้อด้อยเรื่องสัญญาณรบกวนที่เกิดจากตัวฟิโนโตไดโอด ซึ่งปริมาณของสัญญาณรบกวนที่เกิดขึ้นจะแปรผันตามขนาดพื้นที่ของตัวฟิโนโตไดโอด และอุณหภูมิที่ใช้งาน สำหรับวิธีการที่จะลดสัญญาณรบกวนที่เกิดจากอุณหภูมิใช้งานของตัวฟิโนโต

ไดโอดลงได้นั้นต้องใช้ระบบทำความเย็นมาช่วยลดอุณหภูมิของตัวพินโฟโตไดโอด โดยความเย็นที่นิยมใช้กันส่วนใหญ่คือระบบทำความเย็นแบบเทอร์โมอิเล็กทริกคูล (Thermoelectric cool)



ภาพที่ 2.2 สัญลักษณ์และโครงสร้างของซิลิกอนพินโฟโตไดโอด

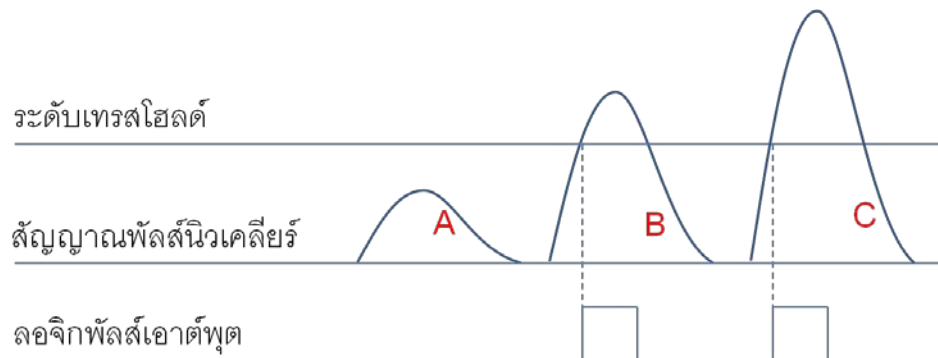


ภาพที่ 2.3 ย่านความยาวคลื่นของแสงที่ถูกปลดปล่อยออกมาจากผลึก CsI(Tl)

2.5.2 วงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดียว [5]

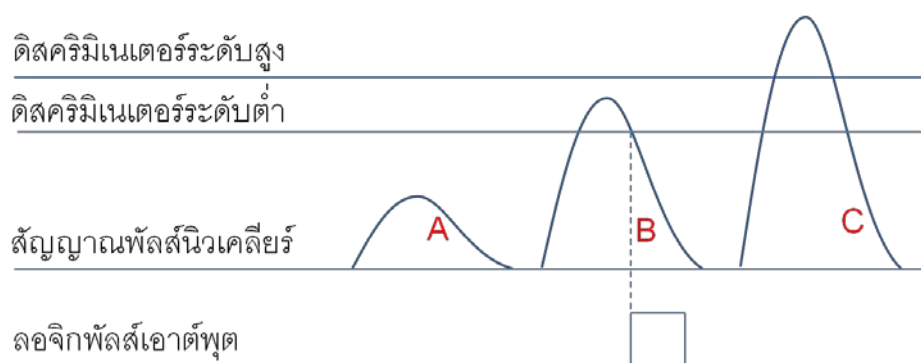
การทำงานของวงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดียว (Single channel analyzer: SCA) เป็นการเลือกวัดค่าพลังงานรังสีเฉพาะระดับความสูงของสัญญาณพัลส์นิวเคลียร์ที่ต้องการ ซึ่งขนาดความสูงของสัญญาณพัลส์นิวเคลียร์จะเป็นค่าที่แสดงถึงค่าของพลังงานรังสีที่เข้ามาตกกระทบกับหัววัดรังสี โดยทั่วไปการเลือกขนาดของสัญญาณพัลส์นิวเคลียร์ที่ได้จากเอาต์พุตของวงจรมหาสัญญาณพัลส์ สามารถทำได้โดยการใช้วงจรที่เรียกว่าดิสคริมิเนเตอร์ (Discriminator) ซึ่งการทำงานของดิสคริมิเนเตอร์ จะให้ลอจิกพัลส์ (Logic pulse) ที่เอาต์พุตเมื่อสัญญาณพัลส์นิวเคลียร์ ที่รับเข้ามามีขนาดความสูงของสัญญาณมากกว่าระดับของเทรชโฮลด์

(Threshold level) ที่ถูกกำหนดค่าไว้จากดิสคริเมเนเตอร์ ดังแสดงในภาพที่ 2.4 จะเห็นได้ว่าขนาดสัญญาณพัลส์นิวเคลียร์ที่ต่ำกว่าระดับเทรชโฮลด์จะไม่เกิดลอจิกพัลส์ ส่วนสัญญาณที่สูงกว่าระดับเทรชโฮลด์จะเกิดลอจิกพัลส์ที่เอาต์พุตของดิสคริเมเนเตอร์ ลักษณะการเกิดลอจิกพัลส์เกิดขึ้นที่ขอบขาขึ้น (Leading edge) ของสัญญาณพัลส์นิวเคลียร์ ณ จุดตัดของสัญญาณพัลส์นิวเคลียร์กับระดับเทรชโฮลด์



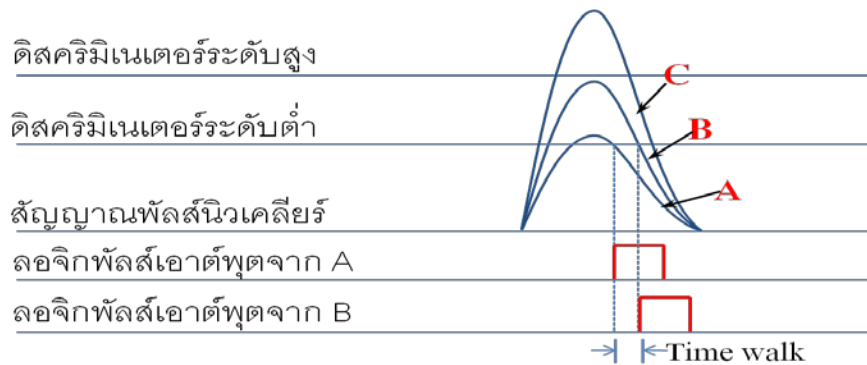
ภาพที่ 2.4 แผนภาพการเกิดสัญญาณลอจิกพัลส์ของอินทิกรัลดิสคริเมเนเตอร์

ในส่วนของการวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดี่ยว ประกอบด้วยดิสคริเมเนเตอร์ 2 ระดับ คือ ดิสคริเมเนเตอร์ระดับล่าง (Lower level discriminator: LLD) และดิสคริเมเนเตอร์ระดับบน (Upper level discriminator: ULD) การทำงานของวงจรวเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดี่ยว จะให้ลอจิกพัลส์ต่อเมื่อสัญญาณพัลส์นิวเคลียร์ที่รับเข้ามีระดับความสูงอยู่ในช่วงระหว่างดิสคริเมเนเตอร์ทั้ง 2 ระดับ ซึ่งเรียกว่าวินโดว์ (ΔE) โดยการเกิดลอจิกพัลส์จะเกิดบริเวณขอบขาลง (Falling edge) ของสัญญาณพัลส์นิวเคลียร์ที่ติดกับระดับของดิสคริเมเนเตอร์ระดับล่าง ดังแสดงในภาพที่ 2.5 ซึ่งจากรูปแสดงให้เห็นว่าสัญญาณพัลส์นิวเคลียร์ B จะถูกสร้างเป็นสัญญาณลอจิกพัลส์

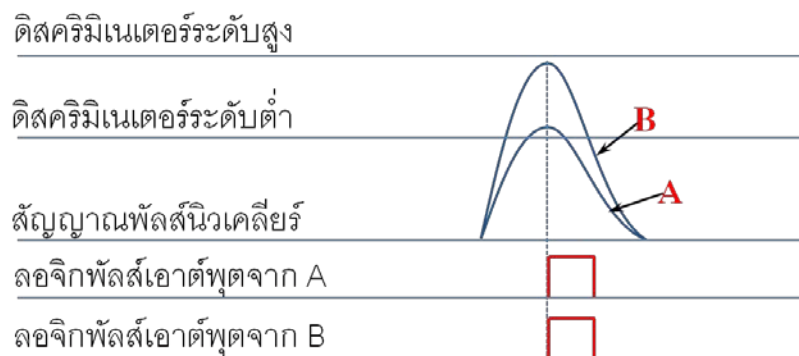


ภาพที่ 2.5 แผนภาพการเกิดสัญญาณลอจิกพัลส์ที่เอาต์พุตของ SCA

วงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดี่ยวสามารถจำแนกตามลักษณะพื้นฐานการทำงานได้ 2 ชนิด คือ นอนไทม์มิง SCA (Non-timing SCA) และ ไทม์มิง SCA (Timing SCA) ในการทำงานแบบนอนไทม์มิง SCA นี้จะให้สัญญาณลอจิกพัลส์ต่อเมื่อสัญญาณพัลส์นิวเคลียร์เกิดขึ้นในวินโดว์ ($\Delta E = ULD-LLD$) โดยสัญญาณลอจิกพัลส์จะเกิดขึ้นที่บริเวณขอบขาลงของสัญญาณพัลส์นิวเคลียร์ที่ตัดกับดิสคริเมเตอร์ระดับล่าง ดังแสดงในภาพที่ 2.6 ซึ่งการทำงานรูปแบบนี้จะส่งผลให้เกิดไทม์วอล์ค ส่วนการทำงานแบบไทม์มิง SCA (Timing SCA) จะมีสัญญาณลอจิกพัลส์เมื่อสัญญาณพัลส์นิวเคลียร์อยู่ในวินโดว์ เหมือนกับแบบนอนไทม์มิง SCA แต่สัญญาณลอจิกพัลส์จะเกิดที่ตำแหน่งเฉพาะดังแสดงในภาพที่ 2.7 เช่น จุดสูงสุดของสัญญาณพัลส์ (Peak sensing) จุดตัดแกนศูนย์ (Zero crossover) หรือแฟรกชันที่กำหนด (Constant fraction) การทำงานแบบไทม์มิง SCA จึงเหมาะกับระบบวิเคราะห์สเปกตรัมเวลา



ภาพที่ 2.6 แผนภาพการเกิดสัญญาณลอจิกพัลส์แบบนอนไทม์มิง SCA



ภาพที่ 2.7 แผนภาพการเกิดสัญญาณลอจิกพัลส์แบบไทม์มิง SCA (Peak sensing)

นอกจากนี้ยังสามารถแบ่งโหมดการทำงานของวงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดี่ยวได้ 4 โหมด ได้แก่

1. โหมดอินทิกรัล (Integral mode)

การทำงานของวงจรวัดความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดียว ในโหมดนี้จะปรับตั้งค่าของดิสคริเมเนเตอร์ระดับบนให้เป็นค่าสูงสุด จึงทำให้มีลักษณะการทำงาน คล้ายคลึงกับการทำงานของวงจรวัดอินทิกรัลดิสคริเมเนเตอร์ดังที่ได้กล่าวมาแล้วในข้างต้น แตกต่าง กันที่รูปแบบการเกิดลอจิกพัลส์ที่เอาต์พุตของวงจรวัดความสูงของสัญญาณพัลส์นิวเคลียร์ แบบช่องเดียวที่จะเกิดที่ขอบขาสูงของสัญญาณพัลส์นิวเคลียร์ ส่วนวงจรวัดอินทิกรัลดิสคริเมเนเตอร์ จะเกิดสัญญาณลอจิกพัลส์ที่ขอบขาขึ้นของสัญญาณพัลส์นิวเคลียร์

2. โหมดปกติ (Normal mode)

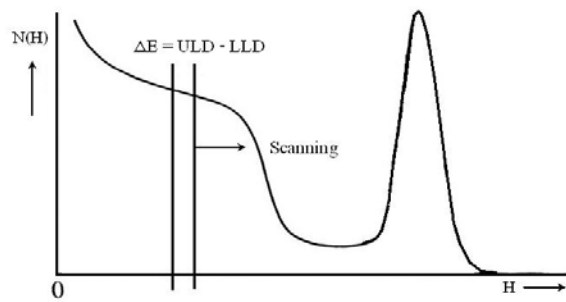
การทำงานของวงจรวัดความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดียว ในโหมดนี้ มีการปรับตั้งค่าของดิสคริเมเนเตอร์ระดับล่าง และดิสคริเมเนเตอร์ระดับบนได้อิสระจาก กัน จึงทำให้สามารถเลือกกำหนดค่าการวิเคราะห์พลังงานได้อย่างเหมาะสมกับช่วงพลังงาน เฉพาะที่สนใจและต้องการตรวจวิเคราะห์

3. โหมดวินโดว์ (Window mode)

การทำงานของวงจรวัดความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดียว ในโหมดนี้ การปรับตั้งค่าของดิสคริเมเนเตอร์ระดับล่างและดิสคริเมเนเตอร์ระดับบน ไม่สามารถทำ ได้อย่างอิสระ โดยทั่วไปการใช้งานต้องตั้งค่าความต่างระหว่างดิสคริเมเนเตอร์ระดับล่างและดิสคริเมเนเตอร์ระดับบนให้คงที่เพื่อใช้เป็นวินโดว์ (ΔE) ซึ่งในการตั้งค่าวินโดว์ต้องสัมพันธ์และเหมาะสม กับความสูงของสัญญาณพัลส์นิวเคลียร์หรือระดับพลังงานของรังสีด้านอินพุตที่ต้องการจะทำการ ตรวจวัด ในส่วนของขั้นตอนการเปลี่ยนค่าดิสคริเมเนเตอร์ระดับล่าง สามารถปรับจากระดับต่ำไป ยังระดับสูงได้อย่างต่อเนื่อง แต่ทุกครั้งของการปรับค่าดิสคริเมเนเตอร์ระดับล่าง ค่าของดิสคริเมเนเตอร์ระดับบนจะเป็นผลรวมของดิสคริเมเนเตอร์ระดับล่างกับวินโดว์

4. โหมดสแกน (Scanning mode)

การทำงานของวงจรวัดความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดียว ในโหมดนี้ มีการทำงานคล้ายกับโหมดวินโดว์ แต่สามารถควบคุมดิสคริเมเนเตอร์ระดับล่างโดยการ รับสัญญาณสแกน (Scanning) จากอุปกรณ์กำเนิดสัญญาณภายนอกและส่งสัญญาณลอจิกพัลส์ ไปบันทึกยังอุปกรณ์นับในแต่ละระดับของวินโดว์อย่างอัตโนมัติและต่อเนื่อง ซึ่งค่าของขนาดวินโดว์ จะต้องคงที่เสมอเมื่อมีการปรับสัญญาณดิสคริเมเนเตอร์ระดับล่าง เหมาะสำหรับการใช้เป็นวงจรวัดวิเคราะห์สเปกตรัมพลังงานของรังสี ดังภาพที่ 2.8

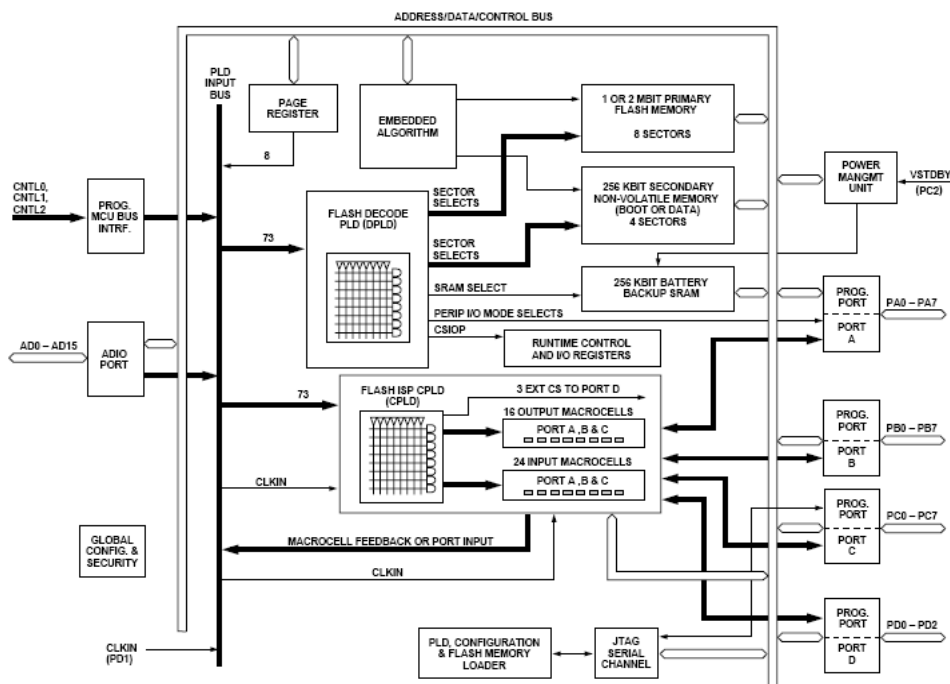


ภาพที่ 2.8 แผนภาพการทำงานของ SCA โหมดสแกน

2.6 ส่วนเชื่อมโยงสัญญาณ

2.6.1 โปรแกรมเมเบิลซิสเต็มดีไวซ์ (พีเอสดี) [6, 7, 8]

โปรแกรมเมเบิลซิสเต็มดีไวซ์หรือไอซี PSD813F2 เป็นไอซีที่มีความเหมาะสม สำหรับเชื่อมโยงสัญญาณกับเครื่องเล่นเกมบอยเพราะสามารถกำหนดฟังก์ชันการทำงานของไอซีเข้ากับไมโครโปรเซสเซอร์ภายในของเครื่องเล่นเกมบอยได้อย่างมีประสิทธิภาพ โครงสร้างภายในของไอซี PSD813F2 ดังแสดงในภาพที่ 2.9 ซึ่งโครงสร้างภายในของไอซี PSD813F2 ประกอบด้วย



ภาพที่ 2.9 แผนภาพโครงสร้างภายในของไอซี PSD813F2

2.6.1.1 หน่วยความจำ

หน่วยความจำภายในของไอซี PSD813F2 ประกอบด้วยหน่วยความจำหลักชนิดแฟลช (Primary Flash memory) ขนาด 16 กิโลไบต์ จำนวน 8 ชุด และหน่วยความจำรองชนิดแฟลช (Secondary Flash memory) ขนาด 8 กิโลไบต์ จำนวน 4 ชุด และหน่วยความจำชนิดสแตติกแรม (Static RAM) ขนาด 2 กิโลไบต์พร้อมระบบสำรองไฟเลี้ยง (Battery backup) โดยมีตัวถอดรหัสภายในไอซี PSD813F2 เป็นตัวกำหนดการเลือกหน่วยความจำในส่วนต่างๆ ดังแสดงในตารางที่ 2.2 นอกจากนี้หน่วยความจำชนิดต่างๆ ดังกล่าวข้างต้นนั้นยังสามารถทำการโปรแกรมหรือทำการลบข้อมูลได้อย่างอิสระโดยการกำหนดขา Ready/Busy ให้ทำงานเป็นเอาต์พุตด้วยโปรแกรม PSDsoft Express

ตารางที่ 2.2 หน่วยความจำภายในไอซี PSD813F2 และสัญญาณเลือกหน่วยความจำจากตัวถอดรหัสพีแอลดี

Sector Number	Primary Flash memory		Secondary Flash memory		SRAM	
	Sector size (Kbytes)	Sector select signal	Sector size (Kbytes)	Sector select signal	Sector size (Kbytes)	Sector select signal
0	16	FS0	8	CSBOOT0	2	RS0
1	16	FS1	8	CSBOOT1		
2	16	FS2	8	CSBOOT2		
3	16	FS3	8	CSBOOT3		
4	16	FS4				
5	16	FS5				
6	16	FS6				
7	16	FS7				
Total	128	8 sectors	32	4 sectors	2	1 sector

2.6.1.2 โปรแกรมเมเบิลลอจิกดีไวซ์

นอกจากนี้ภายในตัวของไอซี PSD813F2 ยังมีฟังก์ชันในการจัดการด้านลอจิกต่างๆ ทำให้สามารถออกแบบฟังก์ชันทางลอจิกได้โดยอาศัยเครื่องมือพีเอสดีเอเบิล (PSD Abel tool) จากตัวโปรแกรม PSDsoft Express ซึ่งฟังก์ชันทางลอจิกประกอบด้วย 2 ส่วน คือ ตัวถอดรหัสพีแอลดี (Decode Programmable Logic Device: DPLD) และ คอมเพล็กซ์พีแอลดี (Complex

Programmable Logic Device: CPLD) โดยที่ตัว DPLD มีหน้าที่ถอดรหัสสำหรับเลือกอุปกรณ์ภายในเช่น หน่วยความจำ รีจิสเตอร์ต่างๆ และอินพุตหรือเอาต์พุตพอร์ต ส่วน CPLD ใช้สำหรับออกแบบฟังก์ชันทางด้านลอจิกต่างๆ เช่น loadable counters, shift registers, state machines และการเข้ารหัสหรือถอดรหัสลอจิกต่างๆ โดยมีเอาต์พุตขนาด 16 บิต อินพุต 24 บิต และแอนดอาร์เรย์ (AND Array) ที่สามารถเชื่อมโยงสัญญาณกับไอซีภายนอกได้ด้วยสัญญาณการเลือกชิป (chip select) ผ่านทางพอร์ตดี

2.6.1.3 อินพุตและเอาต์พุตพอร์ต

ไอซี PSD813F2 มีจำนวนพอร์ตอินพุตและเอาต์พุตทั้งหมด 4 พอร์ตคือ พอร์ตเอ พอร์ตบี และพอร์ตซี ที่มีขนาด 8 บิต ส่วนพอร์ตดีมีขนาด 3 บิต ซึ่งแต่ละขาของพอร์ตต่างๆ สามารถกำหนดฟังก์ชันการทำงานได้อย่างอิสระ การกำหนดฟังก์ชันการทำงานของแต่ละพอร์ตสามารถทำได้โดยอาศัยการจัดคอนฟิกูเรชันภายในด้วยโปรแกรม PSDsoft Express

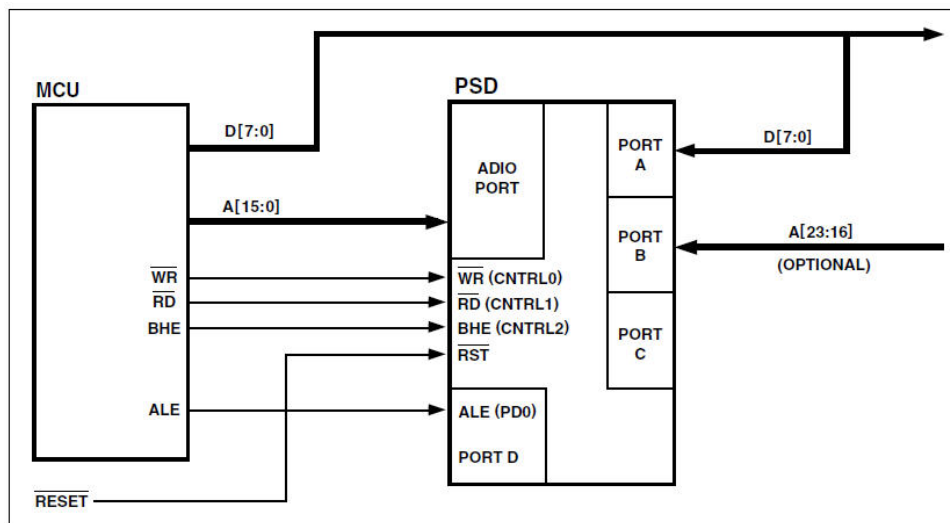
2.6.1.4 การโปรแกรมข้อมูลผ่านทางพอร์ต JTAG

พอร์ต JTAG เป็นส่วนที่เชื่อมต่อระหว่างพอร์ตซีของไอซี PSD813F2 กับพอร์ตนานของเครื่องคอมพิวเตอร์ เพื่อทำการเขียนหรือลบโปรแกรมของฟังก์ชันการทำงานต่างๆ ของไอซี PSD813F2 ที่ออกแบบด้วยโปรแกรม PSDsoft Express

2.6.1.5 ส่วนเชื่อมโยงสัญญาณกับไมโครโพรเซสเซอร์หรือไมโครคอนโทรลเลอร์

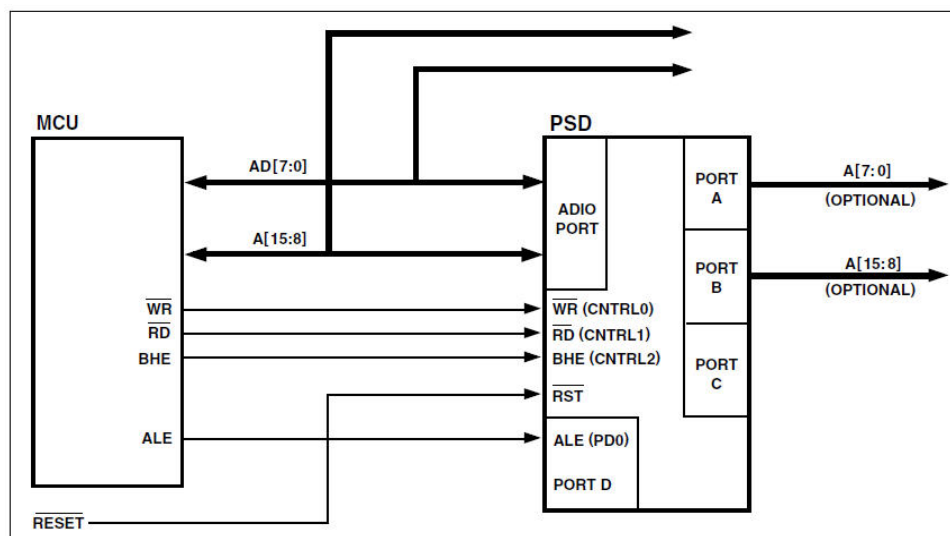
ไอซี PSD813F2 ได้ถูกออกแบบให้มีขีดความสามารถในการเชื่อมโยงสัญญาณกับไมโครโพรเซสเซอร์หรือไมโครคอนโทรลเลอร์ตระกูลต่างๆ ได้ง่ายผ่านทางบัสแอดเดรสและบัสข้อมูล โดยอาศัยการกำหนดรูปแบบการเชื่อมโยงสัญญาณจากโปรแกรม PSDsoft Express ซึ่งในการเชื่อมโยงสัญญาณกับไมโครโพรเซสเซอร์หรือไมโครคอนโทรลเลอร์สามารถทำการเชื่อมโยงได้ทั้ง 2 แบบ คือแบบบัสแอดเดรสชนิดมัลติเพล็กซ์ (Multiplexed bus) และแบบบัสแอดเดรสชนิดไม่มัลติเพล็กซ์ (Non-Multiplexed bus) ขึ้นอยู่กับตระกูลของไมโครโพรเซสเซอร์หรือไมโครคอนโทรลเลอร์ที่นำมาเชื่อมต่อใช้งานโดยมีรายละเอียดดังนี้

1. แบบบัสแอดเดรสชนิดไม่มัลติเพล็กซ์ (Non-Multiplexed Bus) คือการใช้ชุดของบัสข้อมูล (D[7:0]) และชุดของบัสแอดเดรส (A[15:0]) แยกจากกัน ดังแสดงในภาพที่ 2.10 จะเห็นว่าชุดของบัสข้อมูลของไมโครคอนโทรลเลอร์ต่อกับพอร์ตเอของไอซี PSD813F2 และชุดของบัสแอดเดรส ต่อเข้ากับพอร์ตแอดเดรสของไอซี PSD813F2 โดยตรง



ภาพที่ 2.10 แผนภาพการเชื่อมโยงแบบบัสแอดเดรสชนิดไม่มัลติเพล็กซ์

2. แบบบัสแอดเดรสชนิดมัลติเพล็กซ์ (Multiplexed bus) คือการใช้ชุดของ บัสข้อมูล (AD[7:0]) ร่วมกับชุดของบัสแอดเดรส (A[15:8]) ดังแสดงในภาพที่ 2.11 ซึ่งการเลือกชุด สัญญาณทั้ง 2 นั้นเป็นการเลือกจากไมโครคอนโทรลเลอร์ที่นำมาต่อเชื่อมโยงกับไอซี PSD813F2

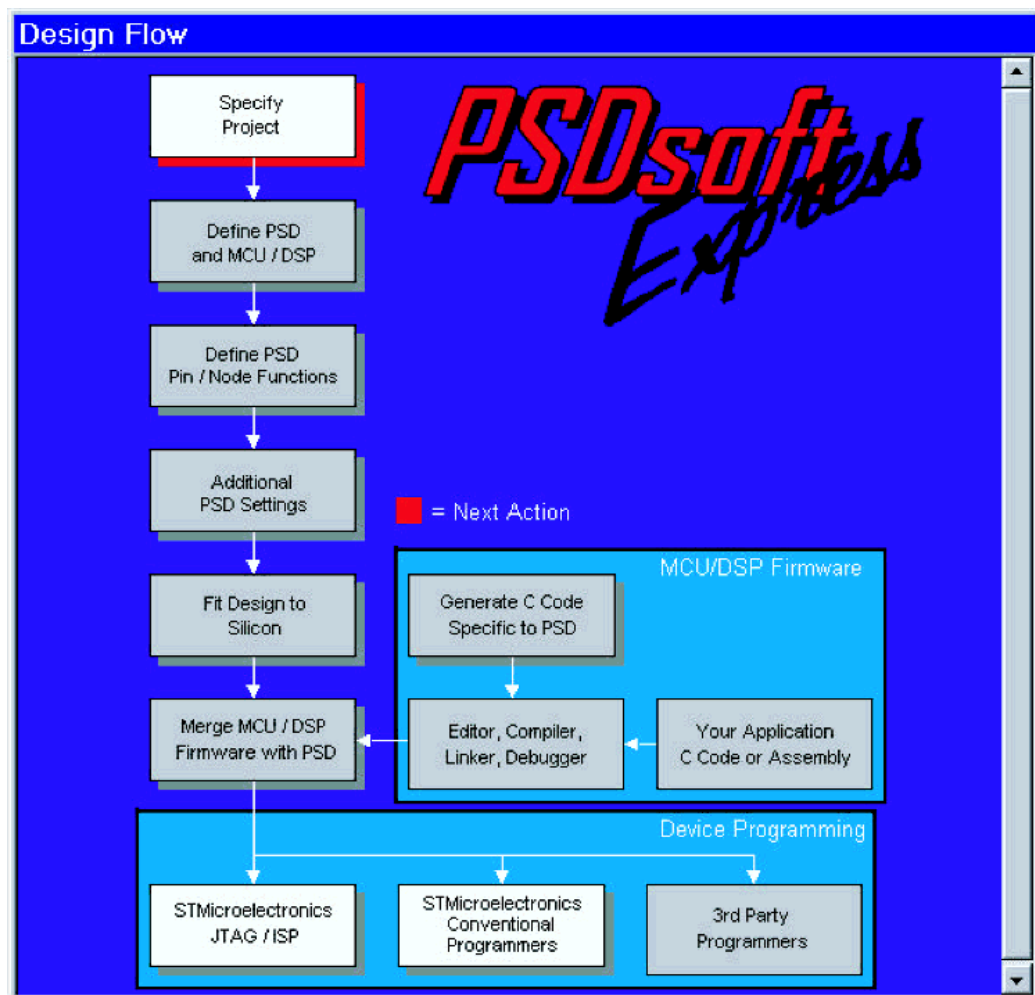


ภาพที่ 2.11 แผนภาพการเชื่อมโยงแบบบัสแอดเดรสชนิดมัลติเพล็กซ์

2.6.2 โปรแกรมพีเอสดีซอฟต์แวร์เอ็กเพรส PSDSoft express program

เนื่องจากการออกแบบฟังก์ชันการทำงานในรูปแบบต่างๆ ของไอซี PSD813F2 สามารถกำหนดโดยการเขียนโปรแกรมด้วยภาษา Advance Boolean Equation Language-Hardware Description Language (ABEL-HDL) ซึ่งเป็นภาษาการอธิบายพฤติกรรมการทำงานทางด้านฮาร์ดแวร์ จึงทำให้ง่ายต่อการใช้งาน ซึ่งในขั้นตอนการออกแบบฟังก์ชันการทำงานต่างๆ เหล่านี้สามารถทำได้โดยโปรแกรม PSDSoft express ดังแสดงในภาพที่ 2.12 โดยมีขั้นตอนหลักของการใช้งานดังนี้

- 1) ใช้ Specify Project สำหรับกำหนดชื่อโปรเจกต์ใหม่
- 2) กำหนด เบอร์ไอซีพีเอสดี และไมโครโพรเซสเซอร์ที่ต้องการนำมาใช้งาน
- 3) กำหนดขาของไอซีพีเอสดี และกำหนดฟังก์ชันต่างๆ



ภาพที่ 2.12 แผนภาพแสดงขั้นตอนการออกแบบฟังก์ชันการทำงาน

เมื่อกำหนดฟังก์ชันการทำงานต่างๆ เสร็จสิ้น จะได้ไฟล์ที่มีนามสกุล .h สำหรับใช้เป็นไฟล์นำเข้า (Include) ในการเขียนโปรแกรมด้วยภาษาซีต่อไป

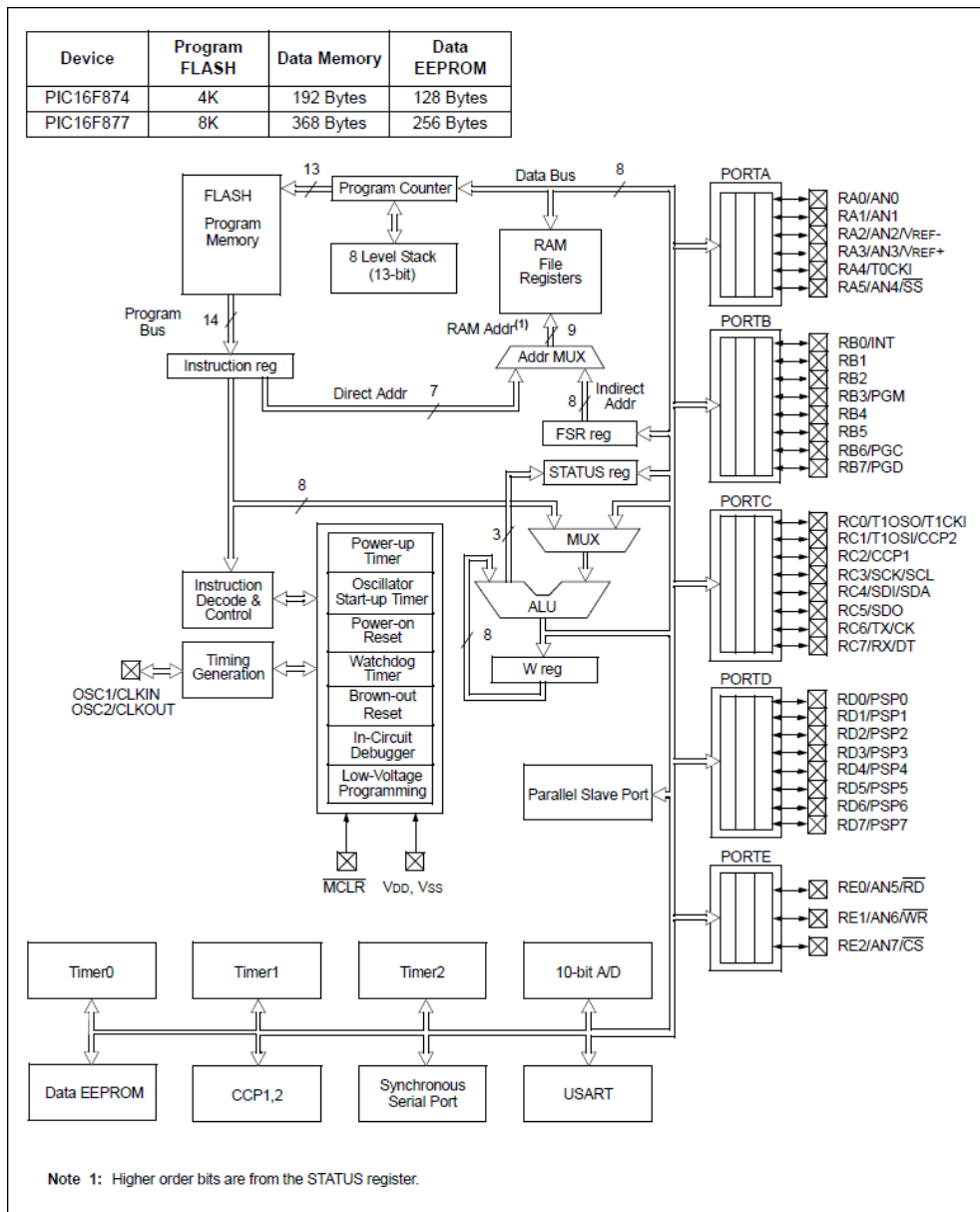
2.6.3 ไมโครคอนโทรลเลอร์ [9]

PIC16F877A เป็นไมโครคอนโทรลเลอร์ที่ได้รับการออกแบบและพัฒนาโดยบริษัทไมโครชิปเทคโนโลยี โดยการทำงานพื้นฐานของไมโครคอนโทรลเลอร์มีการแยกหน่วยความจำโปรแกรม (Program memory) และหน่วยความจำข้อมูล (Data memory) ออกจากกัน มีบัส (Bus) สำหรับติดต่อกับอุปกรณ์ภายนอกขนาด 8 บิต ดังแสดงในภาพที่ 2.13 จากรูปจะเห็นได้ว่าซีพียูที่อยู่ภายในไมโครคอนโทรลเลอร์จะติดต่อกับหน่วยความจำโปรแกรมด้วยบัสแอดเดรสขนาด 14 บิต ที่มีการประมวลผลด้วยคำสั่งขนาด 14 บิตและบัสข้อมูลขนาด 8 บิต สำหรับติดต่อกับหน่วยความจำข้อมูลและรีจิสเตอร์ภายในที่เป็น 8 บิต

คุณสมบัติทางเทคนิคของไมโครคอนโทรลเลอร์ เบอร์ PIC16F877A

- 1) ซีพียูเป็นแบบริสก์ (RISC :Reduce Instruction-Set Computer) มีคำสั่งใช้งาน 35 คำสั่ง
- 2) สามารถกระทำคำสั่งโดยใช้สัญญาณนาฬิกาเพียงหนึ่งลูก ยกเว้นคำสั่งการกระโดด
- 3) ความถี่สูงสุดของสัญญาณนาฬิกาที่ทำงานได้คือ 20 MHz
- 4) มีหน่วยความจำโปรแกรมชนิดแฟลช (FLASH Program Memory) ขนาด 8 กิโลเวิร์ด
- 5) มีหน่วยความจำข้อมูลชนิดแรม (RAM Data Memory) หรือรีจิสเตอร์ขนาด 368 ไบต์
- 6) มีหน่วยความจำข้อมูลอีอีพรอม (EEPROM Data Memory) ขนาด 256 ไบต์
- 7) ตอบสนองแหล่งกำเนิดสัญญาณขัดจังหวะ (Interrupt) ได้ 14 แหล่ง
- 8) มีสแต็ก (Stack) 8 ระดับ
- 9) มีวงจรวินเซอร์ออกรีเซต (POR), เพลเวอร์อัปไทมเมอร์ (PWRT) และอออสซินเลเตอร์อัปไทมเมอร์ (OST)
- 10) มีวงจรวอตช์ด็อกไทมเมอร์ (WDT) ที่มีวงจรรอสซินเลเตอร์ในตัว ทำให้มีความน่าเชื่อถือในการทำงานสูง
- 11) เลือกรักษาข้อมูลทั้งในหน่วยความจำโปรแกรมและหน่วยความจำข้อมูล และเลือกระดับการป้องกันได้

- 12) มีโหมดประหยัดพลังงาน
- 13) สามารถโปรแกรมโดยใช้แรงดัน +5 โวลต์
- 14) แก้ไขข้อมูลในหน่วยความจำโปรแกรมด้วยกระบวนการ ICD (In-circuit Debugger) ผ่านพอร์ตเพียง 2 ขา
- 15) ซีพียูสามารถอ่านและเขียนหน่วยความจำโปรแกรมได้
- 16) ใช้ไฟเลี้ยง +2 โวลต์ ถึง +5.5 โวลต์
- 17) ใช้กระแสซิงก์และซอร์สของพอร์ต 25 มิลลิแอมป์
- 18) ใช้พลังงานไฟฟ้าในกรณีไม่ขับโหลด น้อยกว่า 2 มิลลิแอมป์ ที่ +5 โวลต์ และสัญญาณนาฬิกา 4 MHz, 20 μ A ที่ไฟเลี้ยง +3 V และสัญญาณนาฬิกา 32 kHz น้อยกว่า 1 μ A ในโหมดประหยัดพลังงาน
- 19) มีไทมเมอร์ 3 ตัว คือ ไทมเมอร์ 0 ขนาด 8 บิต มีปริสเกลเลอร์ขนาด 8 บิตในตัว ไทมเมอร์ 1 ขนาด 16 บิต พร้อมปริสเกลเลอร์ และไทมเมอร์ 2 ขนาด 8 บิต มีปริสเกลเลอร์ โพสท์สเกลเลอร์และรีจิสเตอร์คาบเวลา (Period register) ขนาด 8 บิตในตัว
- 20) มีโมดูล CCP 2 ชุด โดยส่วนตรวจจับสัญญาณหรือแคปเจอร์ (Capture) มีขนาด 16 บิต ความละเอียดสูงสุด 12.5 นาโนวินาที (ns) ส่วนเปรียบเทียบสัญญาณ (Compare) มีขนาด 16 บิต ความละเอียดสูงสุด 200 นาโนวินาที วงจร PWM มีความละเอียดสูงสุด 10 บิต
- 21) มีวงจรแปลงสัญญาณอะนาลอกเป็นดิจิทัลขนาด 10 บิต
- 22) มีวงจรเชื่อมต่ออุปกรณ์อนุกรมทั้ง SPI และบัส I²C
- 23) มีวงจรรีเซตข้อมูลอนุกรม (USART) พร้อมการตรวจจับแอดเดรส 9 บิต
- 24) มีวงจรตรวจจับระดับแรงดันไฟเลี้ยง (บราวน์เอาต์ดีเทกชัน: Brown-out detection) เพื่อการรีเซตซีพียู หรือเรียกว่า บราวน์เอาต์รีเซต (Brown-out reset, BOR)



ภาพที่ 2.13 แผนภาพโครงสร้างภายในของไมโครคอนโทรลเลอร์ PIC16F877A [8]

2.7 เครื่องเล่นเกมบอย

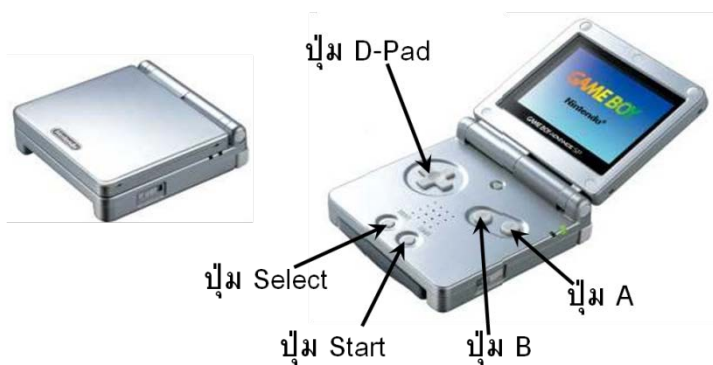
เครื่องเล่นเกมบอยที่ใช้ในงานวิจัยนี้เป็นลิขสิทธิ์ของบริษัทนินเทนโด (NINTENDO) เป็นเครื่องเล่นเกมคอนโซลที่ใช้เพื่อความบันเทิง และเป็นเครื่องเกมที่ได้รับความนิยมอย่างแพร่หลาย เครื่องเล่นเกมบอยรุ่นแรกที่ผลิตคือ เกมบอยพ็อกเก็ต (Gameboy Pocket) หรือที่รู้จักกันในชื่อเกมบอยรุ่นดั้งเดิม ซึ่งได้รับความนิยมอย่างแพร่หลาย แต่ข้อเสียคือส่วนของจอแสดงผลที่เป็นแบบขาวดำ ดังนั้นทางบริษัทนินเทนโดจึงได้ทำการพัฒนารุ่นใหม่ขึ้นมา โดยใช้ชื่อว่า เกมบอยคัลเลอร์ (Gameboy Color) ซึ่งเป็นรุ่นที่ได้รับความนิยมอย่างมากและถือเป็นรุ่นที่ขายดีที่สุด

สำหรับเครื่องเล่นเกมที่มีการผลิตขึ้นมา เนื่องจากจอแสดงผลที่สามารถแสดงสีได้ถึง 56 สี และมีขนาดเล็ก พกพาได้สะดวก หลังจากนั้นได้มีการพัฒนาเครื่องเล่นเกมนุ่นเกมบอยแอดวานซ์ (Gameboy Advance) และนุ่นเกมบอยแอดวานซ์เอสพี (Gameboy Advance SP) ที่มีขนาดของจอแสดงผลที่ 240 x 160 พิกเซล และสามารถแสดงสีได้ถึง 32,768 สี โดยยังคงสามารถเล่นดัลบ์เกมของเกมนุ่นพ็อกเก็ต และเกมนุ่นคัลเลอร์ได้ เพราะมีการใช้ชิพ 2 ชนิด คือ 1) Zilog-Z80 ที่เป็นชิพของเครื่องเล่นเกมนุ่นทั้ง 2 รุ่นที่ได้กล่าวมาข้างต้น และ 2) ARM7TDMI จากนั้นได้พัฒนานุ่นต่อมา คือนุ่น Gameboy Nintendo DS ซึ่งสามารถเล่นได้สองหน้าจอ และสามารถเล่นกับดัลบ์เกมของ Gameboy Advance ได้ด้วย แต่จะแสดงผลเพียงหนึ่งหน้าจอเท่านั้น ซึ่งรายละเอียดของเครื่องเล่นเกมนุ่นแต่ละรุ่นแสดงดังตารางที่ 2.3

ตารางที่ 2.3 รายละเอียดของเครื่องเล่นเกมนุ่นต่าง ๆ

โมเดล ข้อมูลจำเพาะ	Pocket	Color	Advance	Nintendo DS
ชิพ	4.2 MHz 8 Bit Zilog-Z80	8 MHz 8 Bit Zilog-Z80	16.8 MHz 32 Bit ARM7TDMI	67 MHz 32 Bit ARM946E-S
หน่วยความจำ	8 KB	8 KB	32 KB	4 MB
ความละเอียด (พิกเซล)	160x144	160x144	240x160	256x192
สี	2-bit (green to dark green)	15-bit	15-bit	18-bit
ขนาดหน้าจอ (นิ้ว)	2.6	2.6	2.9	3
แบตเตอรี่ (ชั่วโมง)	12	10	15	6-10
น้ำหนัก (กรัม)	394	138	140	280
ยาวxกว้างxสูง (นิ้ว)	5.8x3.5x1.3	5.3x2.9x1.0	3.2x5.7x1.0	3.4x5.9x1.2

งานวิจัยนี้เลือกใช้เครื่องเล่นเกมนุ่นเกมบอยแอดวานซ์เอสพี ซึ่งเป็นรุ่นที่ออกแบบให้มีขนาดกะทัดรัด สามารถพับเก็บได้ ดังแสดงในภาพที่ 2.14 ทำให้สะดวกแก่การพกพา โดยมีคุณสมบัติและรายละเอียดดังตารางที่ 2.4

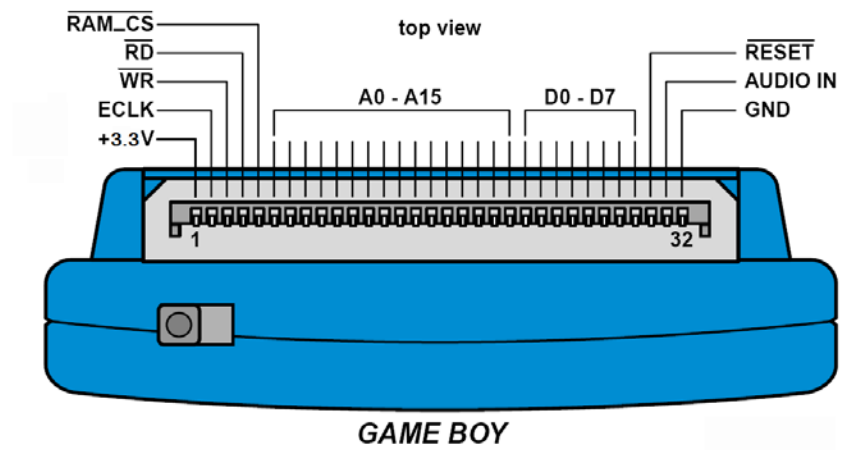


ภาพที่ 2.14 เครื่องเล่นเกมบอยรุ่นเกมบอยแอดวานซ์เอสพี

ตารางที่ 2.4 รายละเอียดของเครื่องเกมบอยรุ่นเกมบอยแอดวานซ์เอสพี

คุณสมบัติ	ข้อมูลจำเพาะ	คุณสมบัติ	ข้อมูลจำเพาะ
ปุ่มควบคุม	D-pad	ความละเอียดของหน้าจอ	240 x 160
น้ำหนัก	140 กรัม	สี	32,768 สี
ขนาด (ยาว x กว้าง x สูง): นิ้ว	3.23 x 3.3 x 0.96	อินพุตและเอาต์พุต	1 x Game Boy Advance link connector, 1 Game Boy Advance SP multi-access port
หน่วยความจำหลัก	64KB/32KB	สัญญาณนาฬิกา	16.78MHz
หน่วยความจำวิดีโอ	96KB	ซีพียู	Sharp ARM7TDMI และ Zilog-Z80
รุ่น	Advance SP	Built-in Features	Front light integrated with LCD
หน้าจอ	2.9 นิ้ว แบบ LCD	แบตเตอรี่	ชนิดลิเทียมไอออน ความจุ 3.7 โวลต์

การเชื่อมโยงสัญญาณกับเครื่องเล่นเกมบอยสามารถเชื่อมโยงสัญญาณแบบขนาน ดังภาพที่ 2.15 แสดงหาสัญญาณที่ใช้ในการเชื่อมโยงกับอุปกรณ์ภายนอก ประกอบด้วยบัสแอดเดรสขนาด 16 บิต บัสข้อมูลขนาด 8 บิต และบัสควบคุมขนาด 5 บิต



ภาพที่ 2.15 ขั้วต่อสัญญาณของเครื่องเล่นเกมบอย

บทที่ 3

วิธีดำเนินการวิจัย

3.1 ข้อมูลพื้นฐานในการออกแบบเครื่องแกมมาสเปกโตรมิเตอร์

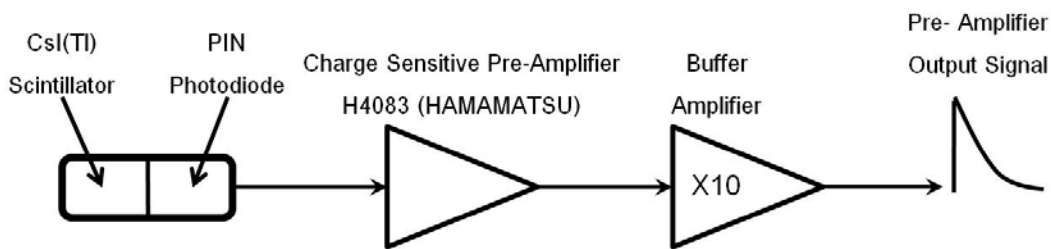
เครื่องแกมมาสเปกโตรมิเตอร์ที่ทำงานร่วมกับเครื่องเล่นเกมบอย ได้ถูกออกแบบให้มีขนาดกะทัดรัด ง่ายต่อการพกพา และเหมาะแก่ศึกษาเรียนรู้เกี่ยวกับเครื่องมือวัดทางด้านรังสี ซึ่งหน้าที่หลักของเครื่องเล่นเกมบอยจะใช้ในการควบคุมการทำงานและเป็นส่วนแสดงผลของเครื่องแกมมาสเปกโตรมิเตอร์ โดยจะทำงานร่วมกับไอซี PSD813F2 สำหรับส่วนของไมโครคอนโทรลเลอร์เบอร์ PIC16F877A ถูกออกแบบให้ใช้ในการเชื่อมโยงสัญญาณระหว่างระบบวัดรังสีแกมมาพร้อมเป็นตัวประมวลผลข้อมูลที่ได้จากการวัดรังสี นอกจากนี้ยังได้ออกแบบให้เครื่องแกมมาสเปกโตรมิเตอร์สามารถเลือกโหมดการใช้งานได้ 2 โหมดดังนี้

1. โหมดสำรวจรังสี (Survey meter) คือโหมดตรวจวัดปริมาณรังสีในบริเวณหรือสถานที่ปฏิบัติงาน แสดงผลในหน่วยของปริมาณรังสีที่ทำให้อากาศแตกตัว (Exposure) ในช่วง 0-100 มิลลิเรินเกนท์ (mR/h) และแสดงผลในหน่วยของจำนวนนับต่อเวลา ในช่วง 0 - 20,000 ครั้งต่อวินาที (CPS)

2. โหมดสเปกตรัมรังสีแกมมา (Gamma spectrometer) คือโหมดที่ใช้วิเคราะห์พลังงานของรังสี และกัมมันตภาพของต้นกำเนิดรังสีจากส่วนที่เป็นโฟโตพีค (Photopeak) สามารถทำงานได้ 2 รูปแบบคือ 1) สแกนสเปกตรัม สามารถกำหนดค่าดิสคริเมเตอร์ระดับล่าง ดิสคริเมเตอร์ระดับบน และวินโดว์พร้อมทั้งกำหนดค่าเวลาในการสแกนสเปกตรัมในแต่ละช่องพลังงานได้ เพื่อใช้สำหรับการเลือกวัดสเปกตรัมเฉพาะพลังงานที่สนใจ และ 2) สวิทช์สเปกตรัม ระบบจะกำหนดค่าดิสคริเมเตอร์ระดับล่าง ดิสคริเมเตอร์ระดับบน และวินโดว์ พร้อมทั้งค่าเวลาในการสแกนสเปกตรัมในแต่ละช่องพลังงาน โดยผู้ใช้เพียงแต่กำหนดจำนวนรอบในการสแกนสเปกตรัมเท่านั้น

3.1.1 ข้อมูลพื้นฐานสำหรับหัววัดรังสีและวงจรขยายสัญญาณพัลส์ส่วนหน้า

หัววัดรังสีแกมมาที่ใช้สำหรับงานวิจัยนี้เป็นชนิดซินทิลเลชันซึ่งใช้ผลึกซีเทียมไอโอไดด์เจือด้วยทาลเลียม (CsI(Tl)) คับปลั๊กกับฟินโฟโตไดโอดยี่ห้อ Hamamatsu รุ่น S3590-08 โดยใช้แหล่งจ่ายไฟฟ้าศักดาต้านทาน ± 12 โวลต์ สัญญาณเอาต์พุตที่ได้จากฟินโฟโตไดโอดจะถูกส่งไปยังวงจรขยายสัญญาณพัลส์ส่วนหน้าแบบไวต่อประจุ (Charge Sensitive Pre-Amplifier) ที่มีอัตราขยายสัญญาณ 12 mV/MeV (Si) ของรังสีแกมมา ซึ่งมีแผนภาพการทำงานดังภาพที่ 3.1



ภาพที่ 3.1 แผนภาพการทำงานของหัววัดรังสีและวงจรรขยายสัญญาณส่วนหน้า

3.1.2 ข้อมูลพื้นฐานสำหรับการออกแบบระบบวัดรังสี

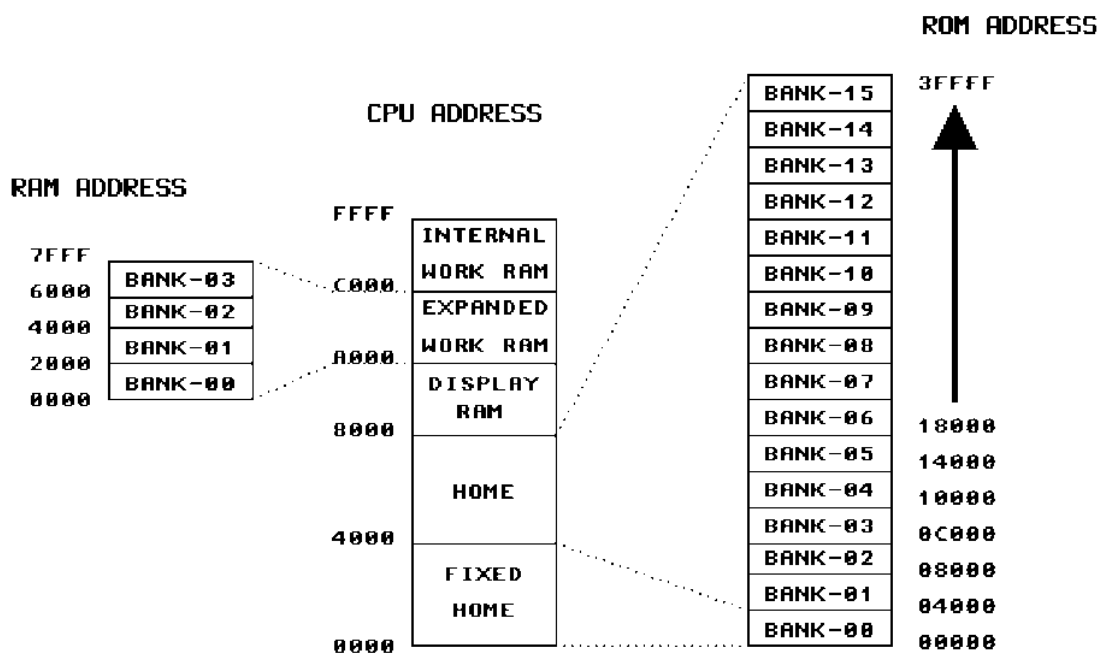
ระบบวัดรังสีประกอบด้วย วงจรแหล่งจ่ายไฟฟ้าคิกดาต้า วงจรรขยายสัญญาณพัลส์ วงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดี่ยว วงจรนับจำนวนพัลส์พร้อมโปรแกรมตั้งเวลา โดยวงจรรขยายสัญญาณพัลส์ทำหน้าที่ขยายสัญญาณที่ได้รับมาจากวงจรรขยายสัญญาณส่วนหน้า เพื่อเพิ่มอัตราส่วนของสัญญาณพัลส์ต่อสัญญาณรบกวน (Signal to Noise Ratio: S/N Ratio) ด้วยการขยายสัญญาณพร้อมทั้งปรับแต่งรูปร่างสัญญาณพัลส์ที่มีขนาดความสูงของสัญญาณพัลส์เป็นสัดส่วนกับระดับพลังงานซึ่งจะถูกวิเคราะห์ด้วยวงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดี่ยว ที่ควบคุมการทำงานด้วยไมโครคอนโทรลเลอร์ สัญญาณลอจิกพัลส์ที่ได้จากวงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดี่ยวในช่วงพลังงานที่กำหนดจะถูกส่งไปยังวงจรรับจำนวนพัลส์พร้อมโปรแกรมตั้งเวลาที่ออกแบบและพัฒนาโปรแกรมภายในไมโครคอนโทรลเลอร์ PIC16F877A

3.1.3 ข้อมูลพื้นฐานสำหรับระบบควบคุมและแสดงผล

การเชื่อมโยงสัญญาณระหว่างระบบวัดรังสีกับเครื่องเล่นเกมบอย ทำได้โดยอาศัยการทำงานร่วมกันของไอซี PSD813F2 และไมโครคอนโทรลเลอร์เบอร์ PIC16F877A ที่ความถี่ 20 MHz โดยทำหน้าที่ในการควบคุมการทำงานของวงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดี่ยว วงจรรับจำนวนพัลส์พร้อมโปรแกรมตั้งเวลา และส่งข้อมูลของจำนวนพัลส์ที่นับได้ไปยังไอซีพีเอสดี เพื่อทำการคำนวณและแปลงค่าจำนวนนับต่อเวลาเป็นอัตราปริมาณรังสีที่วัดได้ เพื่อส่งต่อไปยังเครื่องเล่นเกมบอยสำหรับแสดงผลทางหน้าจอของเครื่องเล่นเกมบอย

การแสดงผลทางหน้าจอของเครื่องเล่นเกมบอยรุ่นเกมบอยแอดวานซ์เอสพี ทำงานด้วย CPU เบอร์ Z80 ของบริษัท Zilog เมื่อนำตลับเกมของเกมบอยคัลเลอร์มาใช้งาน และทำงานด้วย CPU เบอร์ ARM7TDMI ของบริษัท Sharp เมื่อนำตลับเกมของเกมบอยแอดวานซ์มาใช้งาน ซึ่งภายในของเครื่องเล่นเกมบอยมีหน่วยความจำชนิด ROM ขนาด 64 กิโลไบต์ โดยแบ่งออกเป็น

ส่วนของหน่วยความจำสำหรับเก็บโปรแกรม (Code Memory หรือ Program Memory) ขนาด 32 กิโลไบต์ และส่วนของหน่วยความจำสำหรับเก็บข้อมูล (Data Memory) ขนาด 32 กิโลไบต์ แต่เนื่องจากหน่วยความจำภายในมีจำนวนจำกัด การที่จะติดต่อใช้งานร่วมกับเครื่องเล่นเกมบอย จำเป็นต้องเพิ่มหน่วยความจำภายนอก ซึ่งในงานวิจัยนี้เลือกใช้ไอซี PSD813F2 ที่มีหน่วยความจำชนิด Flash ขนาด 16 กิโลไบต์ จำนวน 8 ชุด โดยการจัดสรรหน่วยความจำ (Memory Map) แสดงดังภาพที่ 3.2 เพื่อให้เหมาะสมกับระบบหน่วยความจำภายในของเครื่องเล่นเกมบอย



ภาพที่ 3.2 แผนภาพการจัดสรรหน่วยความจำ (Memory Map) ของเครื่องเล่นเกมบอย

จากภาพที่ 3.2 การจัดสรรระบบหน่วยความจำภายในของเครื่องเล่นเกมบอยสามารถแยกเป็น 5 ส่วน ดังนี้

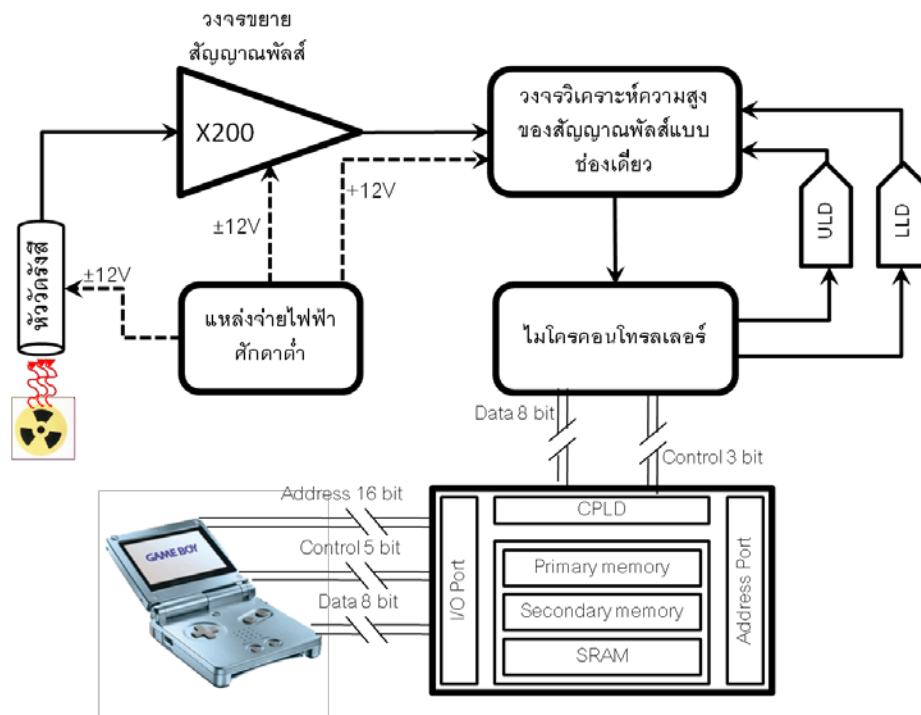
1. Fixed Home เป็นหน่วยความจำที่ใช้เก็บค่าฟังก์ชันการทำงานต่างๆ ของเครื่องเล่นเกมบอย มีขนาด 16 กิโลไบต์ โดยมีตำแหน่งแอดเดรสอยู่ที่ 0000h - 3FFFh
2. Home เป็นหน่วยความจำที่ใช้เก็บข้อมูลโปรแกรมหรือค่าต่างๆ ที่ต้องการนำมาคำนวณ มีขนาด 16 กิโลไบต์ โดยมีตำแหน่งแอดเดรสอยู่ที่ 4000h - 7FFFh ซึ่งในส่วนนี้สามารถเรียกใช้ข้อมูลจากหน่วยความจำภายนอกได้ครั้งละ 1 BANK มีขนาด 16 กิโลไบต์ ดังภาพที่ 3.2

3. Display RAM เป็นหน่วยความจำที่ใช้เก็บข้อมูลในการแสดงผลทางหน้าจอของเครื่องเล่นเกมบอย มีขนาด 8 กิโลไบต์ โดยมีตำแหน่งแอดเดรสอยู่ที่ 8000h – 9FFFh
4. Expanded Work RAM เป็นหน่วยความจำที่ใช้สำหรับเชื่อมต่อกับหน่วยความจำชั่วคราวภายนอก มีขนาด 8 กิโลไบต์ โดยมีตำแหน่งแอดเดรสอยู่ที่ A000h – BFFFh
5. Internal Work RAM เป็นหน่วยความจำชั่วคราวภายใน สำหรับเก็บคำสั่งต่างๆ เกี่ยวกับการควบคุมการทำงานของเครื่องเล่นเกมบอย มีขนาด 16 กิโลไบต์ โดยมีตำแหน่งแอดเดรสอยู่ที่ C000h – FFFFh

ในส่วนของ ROM Address จากภาพที่ 3.2 จะเห็นได้ว่าการแบ่งหน่วยความจำออกเป็น 16 ชุด ชุดละ 16 กิโลไบต์ ซึ่งเท่ากับจำนวนหน่วยความจำในส่วนของ Fixed Home และ Home กล่าวคือไมโครโพรเซสเซอร์สามารถเรียกใช้ข้อมูลในส่วนของ Bank01 – Bank15 ได้ครั้งละ 1 Bank เพื่อนำมาประมวลผลในส่วนของ Home

3.2 การออกแบบและพัฒนาเครื่องเกมมาสเปกโตรมิเตอร์

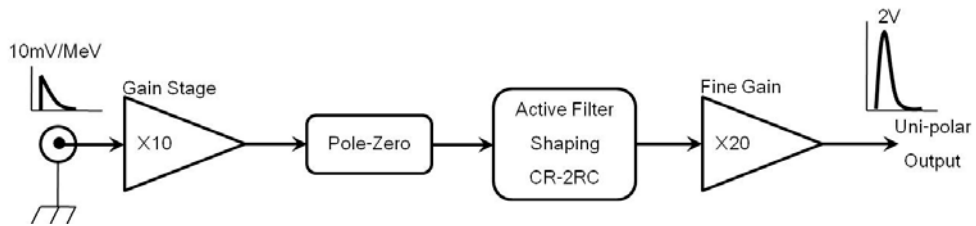
จากแนวคิดในการจัดอุปกรณ์วัดรังสีและควบคุมการทำงานจึงสามารถออกแบบเครื่องเกมมาสเปกโตรมิเตอร์ที่ทำงานร่วมกับเครื่องเล่นเกมบอยแสดงดังภาพที่ 3.3



ภาพที่ 3.3 แผนภาพเครื่องเกมมาสเปกโตรมิเตอร์ที่ทำงานร่วมกับเครื่องเล่นเกมบอย

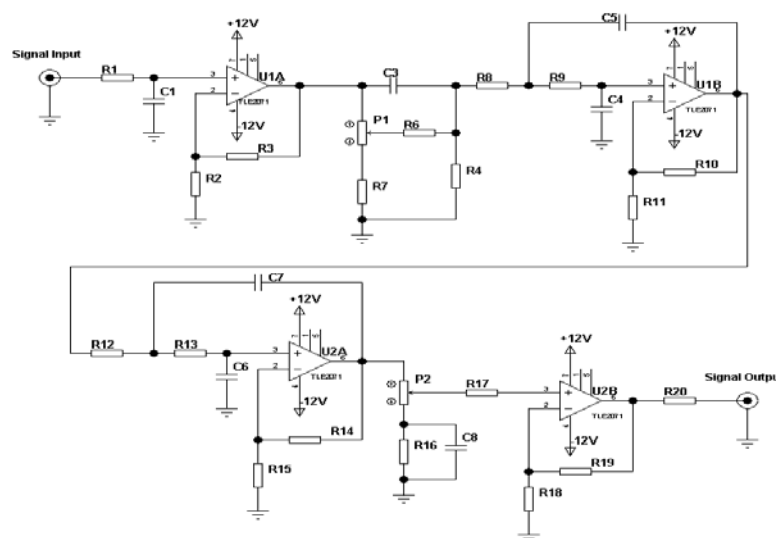
3.2.1 วงจรขยายสัญญาณพัลส์

วงจรขยายสัญญาณพัลส์ทำหน้าที่ขยายสัญญาณและปรับแต่งรูปสัญญาณพัลส์นิวเคลียร์จากวงจรขยายสัญญาณพัลส์ส่วนหน้าให้เหมาะสมกับการนำไปใช้งานยังวงจรวินิจฉัยความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดี่ยวโดยมีแผนภาพการทำงานดังภาพที่ 3.4



ภาพที่ 3.4 แผนภาพการทำงานของวงจขยายสัญญาณพัลส์

จากภาพที่ 3.4 วงจรขยายสัญญาณพัลส์ถูกออกแบบให้มีอัตราขยายสัญญาณรวมเท่ากับ 200 เท่า รวมถึงวงจรลบล้างโพลซีโร (Pole-Zero Cancellation) เพื่อปรับแก้อันเดอร์ชูท (Undershoot) และโอเวอร์ชูท (Overshoot) ของสัญญาณพัลส์นิวเคลียร์เมื่อผ่านวงจรดิฟเฟอเรนเชียล สัญญาณพัลส์นิวเคลียร์ที่ได้จากวงจรลบล้างโพลซีโรจะมีค่าคงตัวเวลาการสลายตัว 3 ไมโครวินาที และปรับแต่งรูปสัญญาณพัลส์นิวเคลียร์ด้วยวงจรกรองความถี่แบบแอคทีฟ (Active Filter Shaping) แบบ Butterworth Low-pass filter (4th Order) ทำให้มีอัตราความสัมพัทธ์ระหว่างสัญญาณพัลส์ต่อสัญญาณรบกวนเท่ากับ 85:1 เท่า ซึ่งสามารถออกแบบวงจรวินิจฉัยสัญญาณได้ดังภาพที่ 3.5 และสร้างแผ่นพิมพ์วงจรดังภาพที่ 3.8

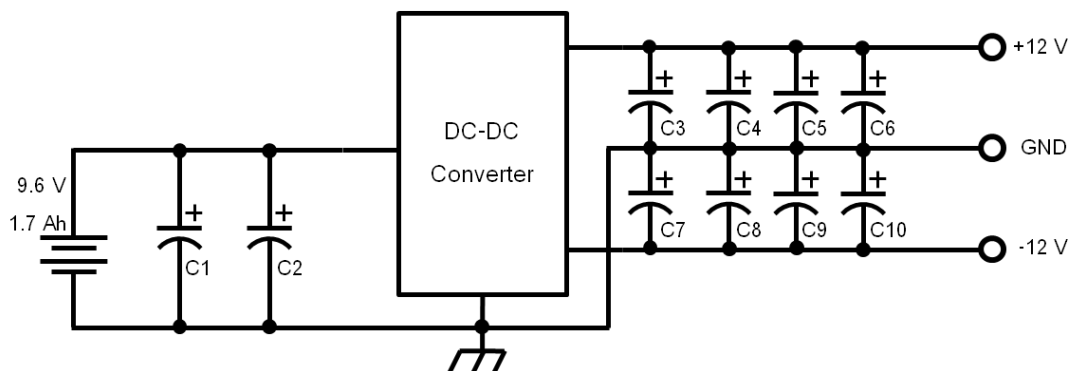


ภาพที่ 3.5 วงจรขยายสัญญาณพัลส์

3.2.2 วงจรแหล่งจ่ายไฟฟ้าคักดาต่ำ

วงจรแหล่งจ่ายไฟฟ้าคักดาต่ำ ได้เลือกใช้โมดูล ดีซี ทู ดีซี คอนเวอร์เตอร์ (DC-DC Converter) ของบริษัท TRACO POWER รุ่น TEL3-2522 สำหรับแปลงค่าของไฟฟ้ากระแสตรง จากแบตเตอรี่แพ็คขนาด 9.6 โวลต์ กระแส 1,700 มิลลิแอมป์ชั่วโมง ให้เป็นไฟฟ้ากระแสตรง ขนาด ± 12 โวลต์ สำหรับใช้งานกับเครื่องเกมมาสเปกโทรมิเตอร์ที่ทำงานร่วมกับเครื่องเล่นเกมบอย โดยมีแผนผังการต่อใช้งานดังแสดงในภาพที่ 3.6 ซึ่งไฟฟ้ากระแสตรงขนาด ± 12 โวลต์จะถูกนำไปใช้กับ ส่วนของวงจรต่างๆ ดังนี้

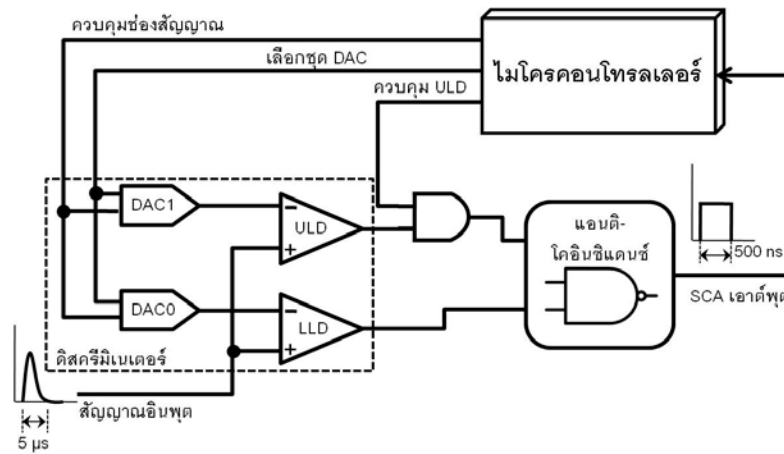
1. คักดาไฟฟ้าขนาด +12 โวลต์ สำหรับจ่ายให้กับ วงจรขยายสัญญาณส่วนหน้า วงจรขยายสัญญาณพัลส์ และแรงดันอ้างอิงสำหรับวงจรวิเคราะห์ความสูงของ สัญญาณพัลส์นิวเคลียร์แบบช่องเดียว
2. คักดาไฟฟ้าขนาด -12 โวลต์ สำหรับจ่ายให้กับ หัววัดรังสีชนิดผลึกซินทิลเลเตอร์ CsI(Tl) วงจรขยายสัญญาณส่วนหน้า และวงจรขยายสัญญาณพัลส์



ภาพที่ 3.6 วงจรแหล่งจ่ายไฟฟ้าคักดาต่ำ

3.2.3 วงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดียว

วงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดียว ทำหน้าที่วิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์ที่ได้จากวงจรขยายสัญญาณพัลส์และเปลี่ยนรูปสัญญาณพัลส์นิวเคลียร์เป็นสัญญาณลอจิกที่มีความกว้าง 500 นาโนวินาที เพื่อส่งต่อไปกับโปรแกรมนับจำนวนสัญญาณพัลส์นิวเคลียร์ที่พัฒนาโปรแกรมภายในไมโครคอนโทรลเลอร์เบอร์ PIC16F877A ซึ่งมีแผนภาพการทำงานของวงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดียวดังภาพที่ 3.7



ภาพที่ 3.7 แผนภาพการทำงานของ SCA

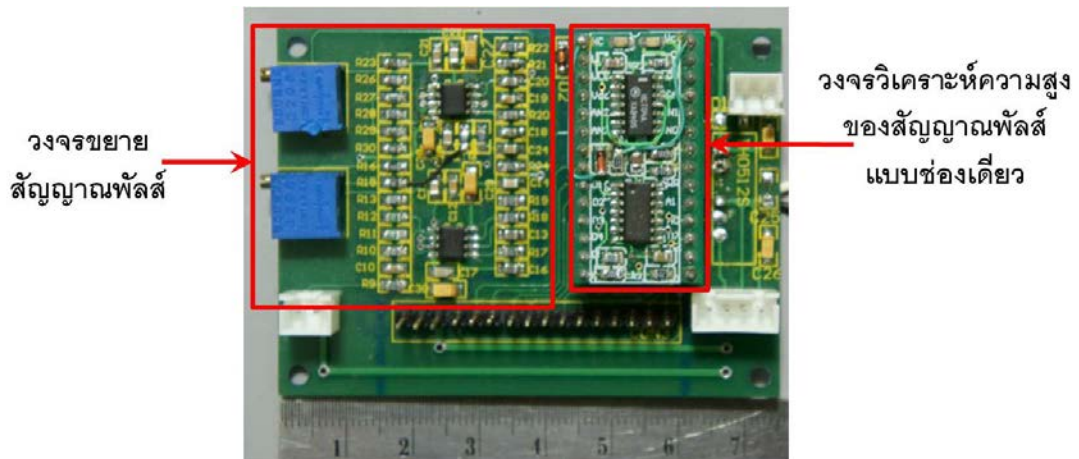
จากภาพที่ 3.7 ออกแบบให้วงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดียวสามารถทำงานได้ 4 โหมด คือ โหมดอินทิกรัล โหมดปกติ โหมดวินโดว์ และโหมดสแกน ตามความเหมาะสมในการใช้งาน โดยมีส่วนประกอบหลัก 2 ส่วน

1) วงจรดิสคริมีเนเตอร์ชนิดโปรแกรมได้ 2 ระดับ (DAC0 และ DAC1) สามารถโปรแกรมค่าศักดาไฟฟ้า 0 ถึง 5 โวลต์ โดยรับสัญญาณควบคุมดิสคริมีเนเตอร์ระดับล่างและดิสคริมีเนเตอร์ระดับบน ขนาด 8 บิต จากไมโครคอนโทรลเลอร์ ทำให้สามารถสร้างช่องวิเคราะห์ความสูงของสัญญาณพัลส์ได้ 256 ช่องสัญญาณ และกำหนดค่าศักดาไฟฟ้าอ้างอิงเท่ากับ 5 โวลต์ ดังนั้นความละเอียดสำหรับช่องวิเคราะห์ความสูงของสัญญาณพัลส์สามารถปรับได้ต่ำสุด 20 มิลลิโวลต์

2) วงจรแอนติโคอิดินซิแดนซ์ โดยการใช้ตัวดำเนินการทางตรรกะ (Logical operator) ชนิดแนนด์ (NAND Gate) เพื่อคัดเลือกสัญญาณพัลส์นิวเคลียร์ที่ได้จากวงจรดิสคริมีเนเตอร์ทั้ง 2 ระดับ โดยมีฟังก์ชันการทำงานดังตารางที่ 3.1 ซึ่งวงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดียวที่ประกอบเสร็จเรียบร้อยแล้วแสดงได้ดังภาพที่ 3.8

ตารางที่ 3.1 ฟังก์ชันการทำงานของวงจรแอนติโคอิดินซิแดนซ์

กรณี	ULD	LLD	SCA OUT	รายละเอียด
1	0	0	0	สัญญาณอินพุตมีค่าต่ำกว่าค่า LLD และ ULD
2	0	1	1	สัญญาณอินพุตมีค่าสูงกว่าค่า LLD และต่ำกว่า ULD
3	1	0	0	ไม่สามารถเกิดขึ้นได้
4	1	1	0	สัญญาณอินพุตมีค่าสูงกว่าค่า LLD และ ULD



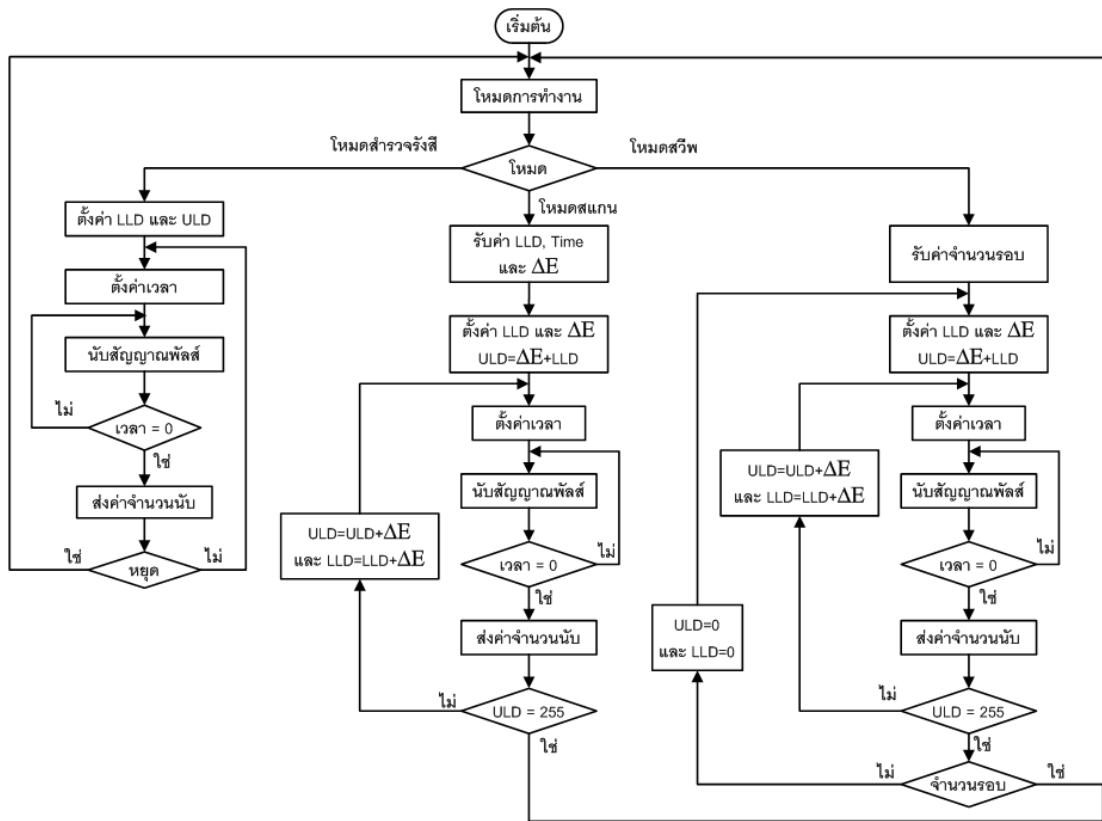
ภาพที่ 3.8 วงจรขยายสัญญาณพัลส์ และวงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดียวที่ประกอบเรียบร้อยแล้ว

3.3 การพัฒนาส่วนเชื่อมโยงสัญญาณและส่วนแสดงผล

เป็นการพัฒนาโปรแกรมควบคุมการทำงานของระบบที่ทำการพัฒนาขึ้น แบ่งออกเป็น 2 ส่วนคือ 1) วงจรนับจำนวนพัลส์พร้อมโปรแกรมตั้งเวลา และ 2) โปรแกรมประมวลผลและแสดงผลบนหน้าจอเครื่องเล่นเกมบอย

3.3.1. วงจรนับจำนวนพัลส์พร้อมโปรแกรมตั้งเวลา

วงจรถับจำนวนพัลส์พร้อมโปรแกรมตั้งเวลาได้ถูกพัฒนาขึ้นโดยอาศัยไมโครคอนโทรลเลอร์เบอร์ PIC16F877A ด้วยภาษาเบสิก สำหรับตั้งเวลาและนับจำนวนสัญญาณพัลส์นิวเคลียร์ ด้วยการส่งสัญญาณควบคุมโหมดการทำงาน และรับสัญญาณพัลส์นิวเคลียร์ที่ได้จากวงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดียว โดยอาศัยฟังก์ชัน Timer0 ในการตั้งเวลา และใช้ Timer1 ในการเก็บค่าจำนวนนับขนาด 16 บิต เพื่อส่งให้ไอซี PSD813F2 ในรูปแบบของข้อมูลแบบขนาน ขนาด 8 บิต จำนวน 2 ครั้ง โดยส่งเป็นค่า High Byte และ Low Byte ซึ่งข้อมูลเหล่านี้จะถูกนำไปใช้สำหรับการคำนวณค่าจำนวนนับและเปลี่ยนให้อยู่ในรูปแบบของหน่วยวัดปริมาณรังสีต่อไป ซึ่งขั้นตอนการทำงานของโปรแกรมที่ได้ทำการพัฒนาแสดงดังแผนผังในภาพที่ 3.9

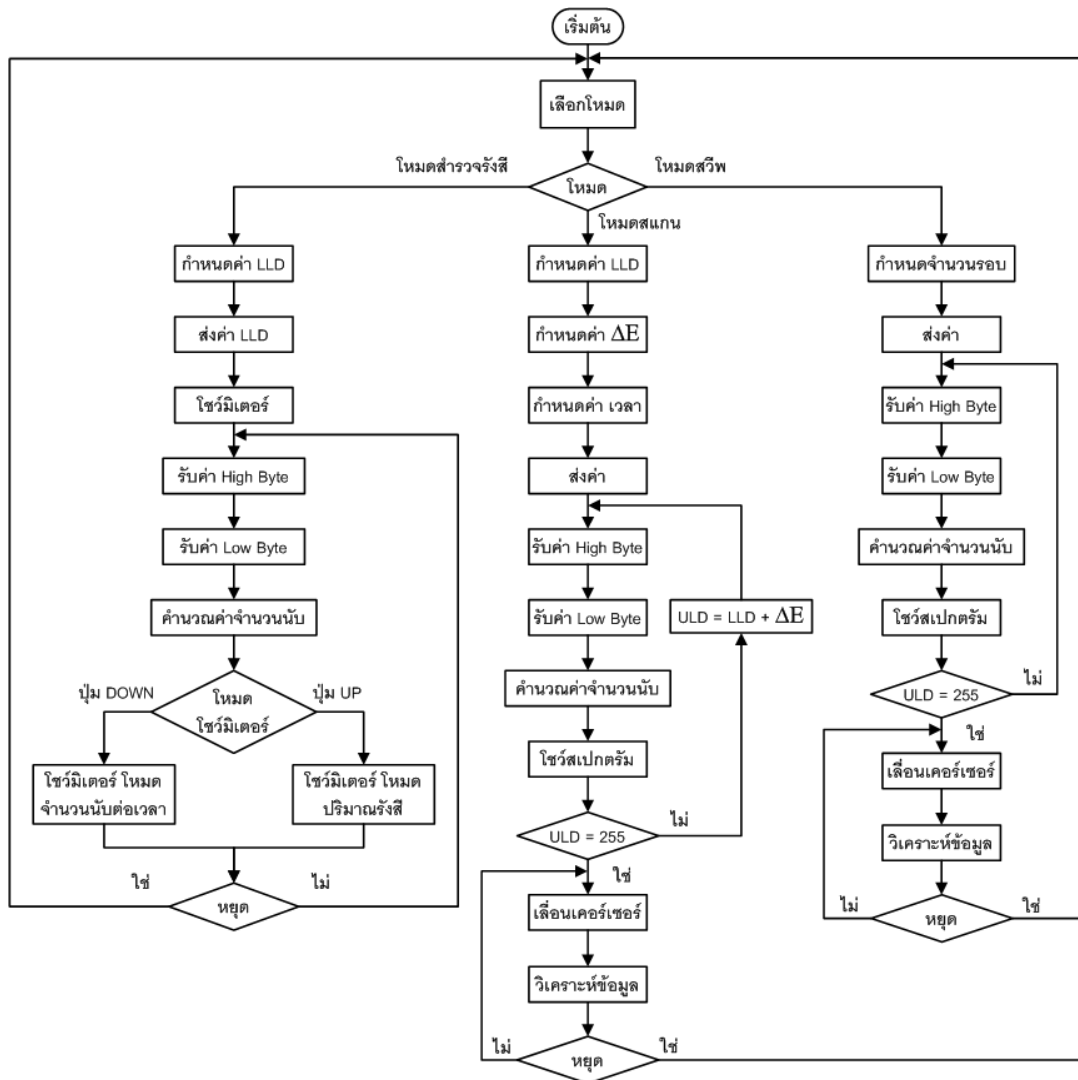


ภาพที่ 3.9 แผนผังการทำงานของวงจรมับจำนวนพัลส์พร้อมโปรแกรมตั้งเวลา

จากภาพที่ 3.9 โปรแกรมเริ่มต้นการทำงานด้วยการรับสัญญาณเลือกโหมดการทำงานของวงจรวเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดี่ยวจากไอซี PSD813F2 โดยออกแบบโปรแกรมให้สามารถทำงานได้ 3 โหมด คือโหมดอินทิกรัล โหมดสแกนสเปกตรัมและโหมดสวิตช์สเปกตรัม

3.3.2. โปรแกรมประมวลผลและแสดงผลบนหน้าจอของเครื่องเล่นเกมบอย

โปรแกรมประมวลผลและแสดงผลบนหน้าจอของเครื่องเล่นเกมบอยออกแบบโดยพัฒนาโปรแกรมบนไอซี PSD813F2 ด้วยภาษา Advance Boolean Equation Language-Hardware Description Language (ABEL-HDL) ซึ่งเป็นภาษาอธิบายพฤติกรรมการทำงานทางด้านฮาร์ดแวร์ สำหรับกำหนดฟังก์ชันการทำงานต่างๆ ของระบบ และพัฒนาโปรแกรมาร่วมกับภาษาซี สำหรับเรียกใช้ฟังก์ชันต่างๆ ที่ได้ออกแบบไว้เพื่อเชื่อมโยงสัญญาณกับไมโครคอนโทรลเลอร์ภายในของเครื่องเล่นเกมบอยและเชื่อมโยงสัญญาณกับไมโครคอนโทรลเลอร์ PIC16F877A พร้อมทั้งพัฒนาโปรแกรมสำหรับเลือกโหมดการทำงานและแสดงผลของระบบ โดยแบ่งออกเป็น 3 โหมด คือ โหมดสำรวจจริงสี่ โหมดสแกนสเปกตรัมและโหมดสวิตช์สเปกตรัม ซึ่งมีแผนผังการทำงานดังภาพที่ 3.10



ภาพที่ 3.10 แผนผังการทำงานของระบบประมวลผลและแสดงผลบนเครื่องเล่นเกมบอย

จากภาพที่ 3.10 แสดงแผนภาพการทำงานของระบบประมวลผลและแสดงผลบนเครื่องเล่นเกมบอย เมื่อโปรแกรมเริ่มทำงานจะแสดงหน้าจอสำหรับเลือกโหมตการใช้งานของระบบ ซึ่งสามารถเลือกโหมตการทำงานโดยการใช้นิ้ว (Up หรือ Down) ของเครื่องเล่นเกมบอย สำหรับเลือกโหมต จากรูปจะเห็นได้ว่าระบบสามารถเลือกโหมตการทำงานได้ 3 โหมต คือ

1. โหมตสำรวจจริงสี่ ผู้ใช้จะต้องกำหนดค่าโดยการกดปุ่ม Up หรือ Down เพื่อเพิ่มลดค่าดิสคริเมเนเตอร์ระดับล่าง (LLD) สำหรับตัดสัญญาณรบกวน และกดปุ่ม Select สำหรับส่งค่าที่กำหนดให้กับไมโครคอนโทรลเลอร์ PIC16F877A เพื่อตั้งค่าดิสคริเมเนเตอร์ระดับล่างของวงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดี่ยว และรอรับค่าจำนวนนับที่ส่งมาจากไมโครคอนโทรลเลอร์ โดยการรับข้อมูลเป็น 8 บิตบน และ 8 บิตล่าง และนำมาคำนวณโดยใช้สมการที่ 3.1 ในการแปลงค่าข้อมูลจำนวน 8 บิต 2 ชุด ให้เป็น ข้อมูลจำนวน 16 บิต 1 ชุด และใช้

สมการที่ 3.2 ในการแปลงค่าจำนวนนับต่อเวลาเป็นค่าอัตราปริมาณรังสีที่ทำให้อากาศแตกตัว ที่มีหน่วยการวัดเป็น เรินเกนท์ต่อชั่วโมง (R/H)

$$C = (\text{High byte} \times 255) + \text{Low byte} \dots\dots\dots (3.1)$$

โดยที่

C = จำนวนนับที่นับได้ ขนาด 16 บิต หน่วยจำนวนนับต่อวินาที

High byte = จำนวนนับไบต์สูงที่รับจากไมโครคอนโทรลเลอร์ ขนาด 8 บิต

Low byte = จำนวนนับไบต์ต่ำที่รับจากไมโครคอนโทรลเลอร์ ขนาด 8 บิต

$$\dot{X} = kC \dots\dots\dots (3.2)$$

โดยที่

\dot{X} = อัตราปริมาณรังสีที่ทำให้อากาศแตกตัว หน่วย R/h (เรินเกนท์ต่อชั่วโมง)

C = ค่าจำนวนนับต่อวินาที

k = ค่าคงที่ปริมาณรังสีที่ทำให้อากาศแตกตัวของหัววัดรังสี

2. โหมดสแกน ผู้ใช้จะต้องกำหนดค่าโดยการกดปุ่ม Up หรือ Down เพื่อเพิ่มลดดิสคริมิเนเตอร์ระดับล่าง ค่าวินโดว์ และค่าเวลาสำหรับสแกนสเปกตรัมในแต่ละช่อง ซึ่งสามารถกำหนดค่าของวินโดว์ได้ต่ำสุดที่ 20 มิลลิโวลต์ต่อหนึ่งช่องสัญญาณ เมื่อกำหนดค่าต่างๆ แล้วกดปุ่ม Select สำหรับส่งค่าทั้งหมดให้กับไมโครคอนโทรลเลอร์ PIC16F877A เพื่อตั้งค่าให้กับวงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดียว และรอรับค่าจำนวนพัลส์ที่ได้จากวงจรนับจำนวนพัลส์พร้อมโปรแกรมตั้งเวลา เพื่อคำนวณค่าดังสมการที่ 3.1 และแสดงบนหน้าจอเครื่องเล่นเกมบอยในรูปแบบสเปกตรัมพลังงานของรังสี เมื่อสแกนสเปกตรัมเสร็จสิ้นสามารถเลื่อนเคอร์เซอร์โดยการกดปุ่ม Left หรือ Right และสามารถขยายและย่อหน้าจอในแนวตั้งโดยการกดปุ่ม Up หรือ Down สำหรับวิเคราะห์ข้อมูลของสเปกตรัม

3. โหมดสวิตช์ ผู้ใช้จะต้องกำหนดค่าโดยการกดปุ่ม Up หรือ Down เพื่อเพิ่มลดจำนวนรอบในการสวิตช์ และกดปุ่ม Select สำหรับส่งค่าให้กับไมโครคอนโทรลเลอร์ PIC16F877A เพื่อตั้งค่าให้กับวงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดียว และรอรับค่าจำนวนพัลส์ที่ได้จากวงจรนับจำนวนพัลส์พร้อมโปรแกรมตั้งเวลา เพื่อคำนวณค่าดังสมการที่ 3.1 และแสดงบนหน้าจอเครื่องเล่นเกมบอยในรูปแบบสเปกตรัมพลังงานของรังสี เมื่อสแกนสเปกตรัมเสร็จสิ้นสามารถ

เลื่อนเคอร์เซอร์โดยการกดปุ่ม Left หรือ Right และสามารถขยายและย่อหน้าจอในแนวตั้งโดยการกดปุ่ม Up หรือ Down สำหรับวิเคราะห์ข้อมูลของสเปกตรัม

ในการหยุดการทำงานของเครื่องแกมมาสเปกโตรมิเตอร์ที่พัฒนาขึ้นในแต่ละโหมดทำได้โดยการกดปุ่ม B ของเครื่องเล่นเกมบอยคังไว้ประมาณ 1 วินาที ซึ่งจะทำให้โปรแกรมกลับไปเมนูหลัก เพื่อให้ผู้ใช้งานสามารถเลือกโหมดการทำงานและกำหนดค่าต่างๆ ได้ใหม่

บทที่ 4

การทดสอบสมรรถนะของระบบ

เครื่องแกมมาสเปกโตรมิเตอร์ที่ทำงานร่วมกับเครื่องเล่นเกมบอยที่พัฒนาขึ้น ผู้ใช้สามารถควบคุมการทำงานของระบบได้จากปุ่มควบคุมของเครื่องเล่นเกมบอย การทดสอบสมรรถนะการทำงานของวงจรต่างๆ และการทำงานโดยรวมของเครื่องแกมมาสเปกโตรมิเตอร์ ได้แก่

1. การทดสอบวงจรแหล่งจ่ายไฟฟ้้าศักดาต่ำ
2. การทดสอบวงจรขยายสัญญาณพัลส์ส่วนหน้า
3. การทดสอบวงจรขยายสัญญาณพัลส์
4. การทดสอบวงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดียว
5. การทดสอบวงจรมับจำนวนพัลส์พร้อมโปรแกรมตั้งเวลา
6. การทดสอบความเป็นเชิงเส้นในการแปลงสัญญาณพัลส์
7. การทดสอบเครื่องแกมมาสเปกโตรมิเตอร์ โหมดสำรวจจริงสี
8. การทดสอบเครื่องแกมมาสเปกโตรมิเตอร์ โหมดสแกน
9. การทดสอบเครื่องแกมมาสเปกโตรมิเตอร์ โหมดสวีพ

4.1 การทดสอบวงจรแหล่งจ่ายไฟฟ้้าศักดาต่ำ

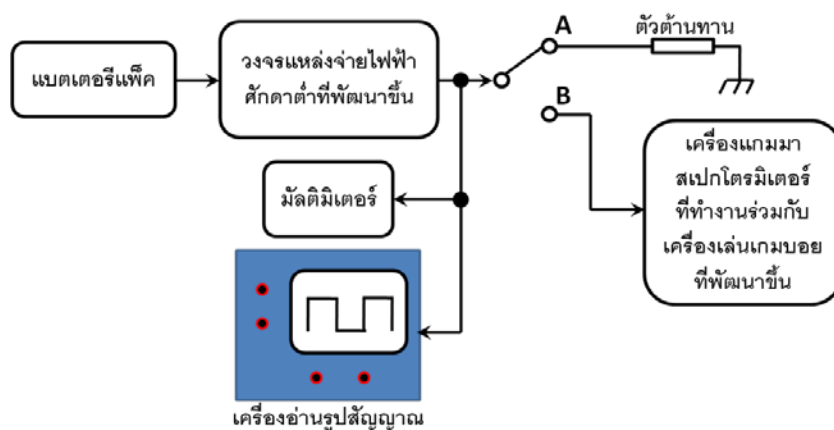
วงจรแหล่งจ่ายไฟฟ้้าศักดาต่ำขนาด ± 12 โวลต์ จากโมดูล DC/DC Converter สามารถจ่ายค่ากระแสสูงสุดให้กับโหลดได้ 125 มิลลิแอมป์ ถูกใช้สำหรับเป็นแหล่งจ่ายให้กับหัววัดรังสี วงจรขยายสัญญาณพัลส์ส่วนหน้า วงจรขยายสัญญาณพัลส์ และแรงดันไฟฟ้าอ้างอิงให้กับวงจร SCA สามารถทดสอบได้ดังนี้

4.1.1 เครื่องมือและอุปกรณ์

1. แบตเตอรี่แพ็ค ขนาด 1.2 โวลต์ กระแส 1,700 มิลลิแอมป์ชั่วโมง จำนวน 8 ก้อน
2. ดิจิทัลมัลติมิเตอร์ยี่ห้อ Sanwa รุ่น PC5000
3. เครื่องอ่านรูปสัญญาณ (Oscilloscope) ของบริษัท Tektronix รุ่น TDS360
4. ตัวต้านทานชนิดเซรามิค
5. วงจรแหล่งจ่ายไฟฟ้้าศักดาต่ำที่นำมาใช้งาน

4.1.2 ขั้นตอนการทดสอบ

1. จัดอุปกรณ์ทดสอบดังภาพที่ 4.1



ภาพที่ 4.1 การจัดอุปกรณ์ทดสอบแหล่งจ่ายไฟฟ้าคิกดาต้า

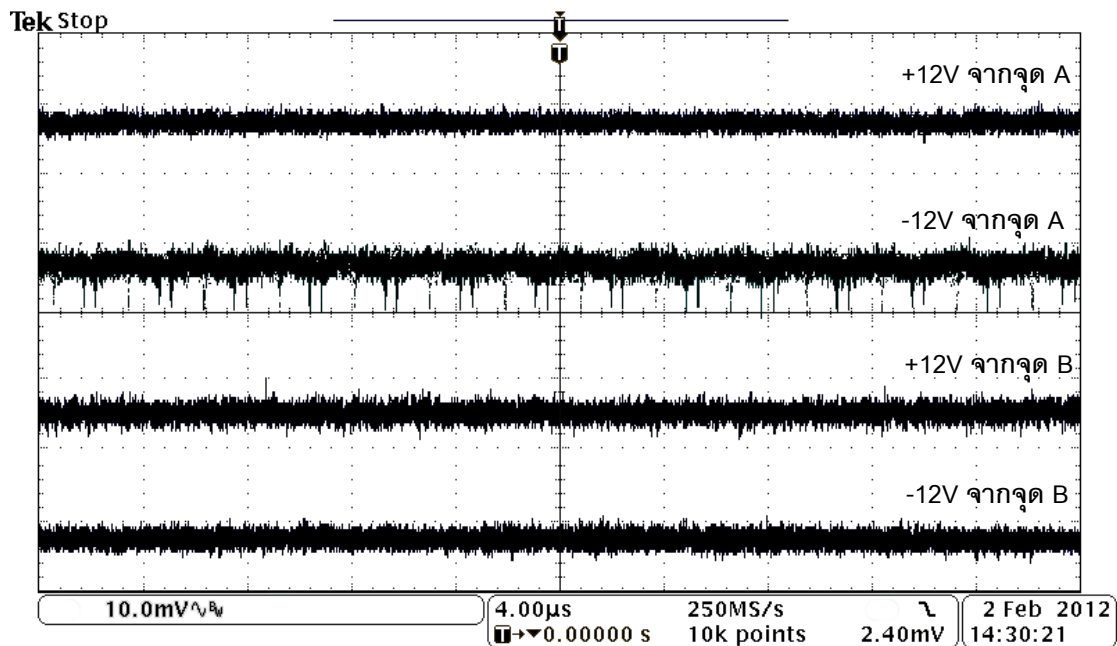
2. ทดสอบแหล่งจ่ายไฟฟ้าคิกดาต้าโดยการใช้ตัวต้านทานชนิดเซรามิกมาทำเป็นโหลด ที่จุด A เพื่อดูแรงดันไฟฟ้า และกระแสสูงสุด ของแหล่งจ่ายไฟฟ้าคิกดาต้าตามสเปคของบริษัทผู้ผลิต และทำการวัดกระแสสูงสุดจากเครื่องเกมมาสเปกโทรมิเตอร์ที่ทำงานร่วมกับเครื่องเล่นเกมบอยที่พัฒนาขึ้น ที่จุด B

4.1.3 ผลการทดสอบ

จากการทดสอบพบว่าแหล่งจ่ายไฟฟ้าคิกดาต้าสามารถจ่ายแรงดันไฟฟ้า และกระแสสูงสุด ดังตารางที่ 4.1 ซึ่งแหล่งจ่ายไฟฟ้าคิกดาต้าแต่ละชุดจะมีค่าสัญญาณรบกวนและริบเปิดสูงสุดประมาณ 10 มิลลิโวลต์ ดังภาพที่ 4.2

ตารางที่ 4.1 อัตราการใช้กำลังไฟฟ้าสูงสุดของแต่ละแหล่งจ่ายไฟฟ้าคิกดาต้า

แหล่งจ่ายไฟฟ้า กระแสตรง	แรงดันไฟฟ้าที่วัดได้ (โวลต์)	กระแสไฟฟ้าที่วัดได้ (มิลลิแอมป์)	กำลังไฟฟ้า (วัตต์)
ทดสอบโดยใช้ตัวต้านทาน 100 โอห์ม ชนิดเซรามิกเป็น โหลด (จุด A)			
ที่ขั้ว -12 โวลต์	11.22	107	1.20
ที่ขั้ว +12 โวลต์	11.18	106	1.19
ทดสอบโดยการใช้เครื่องเกมมาสเปกโทรมิเตอร์ที่พัฒนาขึ้น (จุด B)			
ที่ขั้ว -12 โวลต์	12.01	70	0.84
ที่ขั้ว +12 โวลต์	11.98	45	0.54



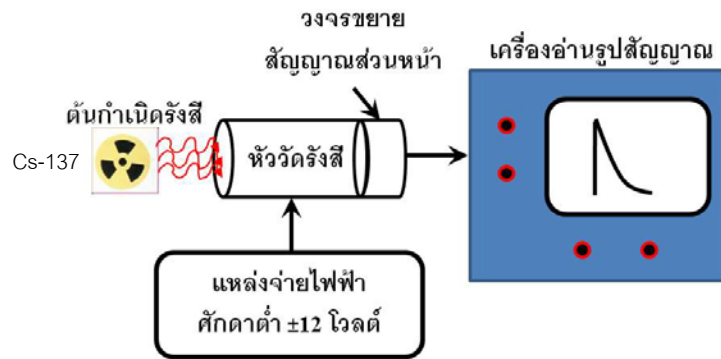
ภาพที่ 4.2 สัญญาณรบกวนและรีปเปิลของวงจรแหล่งจ่ายไฟฟ้าคิกดาต้า

4.2 การทดสอบวงจรขยายสัญญาณพัลส์ส่วนหน้า

การทดสอบวงจรขยายสัญญาณพัลส์ส่วนหน้า ได้ทำการทดสอบรูปสัญญาณ ความสูงของสัญญาณเปรียบเทียบกับค่าพลังงาน (mV/MeV) และเวลาการสลายตัวของสัญญาณพัลส์เอาต์พุต

4.2.1 เครื่องมือและอุปกรณ์

1. ต้นกำเนิดรังสีซีเซียม 137 (Cs-137 : พลังงาน 0.662 MeV)
2. เครื่องอ่านรูปสัญญาณ (Oscilloscope) ของบริษัท Tektronix รุ่น TDS360
3. หัววัดรังสีชนิดซินทิลเลชันที่ทำงานร่วมกับฟิโนไฟโตไดโอด
4. วงจรขยายสัญญาณพัลส์ส่วนหน้า
5. แหล่งจ่ายไฟฟ้าคิกดาต้า ± 12 โวลต์



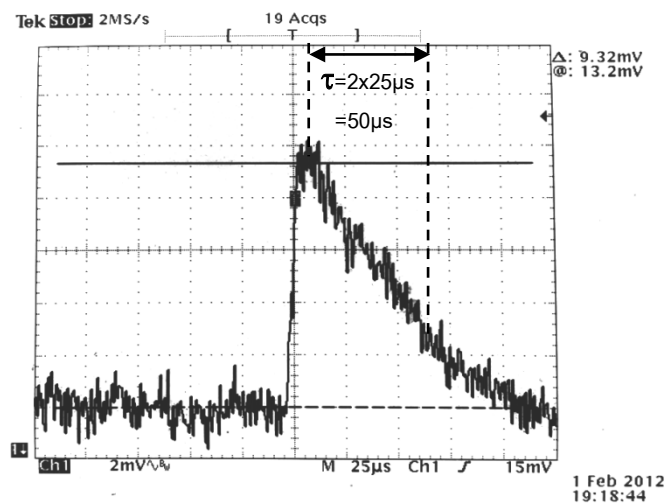
ภาพที่ 4.3 การจัดอุปกรณ์ทดสอบระบบวัดส่วนหน้า

4.2.2 ขั้นตอนการทดสอบ

1. จัดอุปกรณ์ทดสอบดังภาพที่ 4.3
2. ทำการไบอัสหีววัดรังสีชนิดซินทิลเลชันที่ทำงานร่วมกับพินโฟโตไดโอด ด้วยแหล่งจ่ายไฟฟ้าตักดาด่า ± 12 โวลต์ นำต้นกำเนิดรังสีซีเซียม 137 มาวางด้านหน้าหีววัดรังสี อ่านรูปสัญญาณพัลส์ทางเอาต์พุตของวงจรขยายสัญญาณพัลส์ส่วนหน้าด้วยเครื่องอ่านรูปสัญญาณ

4.2.3 ผลการทดสอบ

ภาพที่ 4.4 แสดงสัญญาณเอาต์พุตจากวงจรขยายสัญญาณพัลส์ส่วนหน้าของหีววัดรังสีชนิดซินทิลเลชันที่ทำงานร่วมกับพินโฟโตไดโอด พบว่าขนาดความสูงของสัญญาณพัลส์เท่ากับ 9 มิลลิโวลต์ ที่พลังงาน 0.662 MeV ของต้นกำเนิดรังสีซีเซียม 137 และเวลาการสลายตัวของสัญญาณพัลส์ (τ) เท่ากับ 50 ไมโครวินาที



ภาพที่ 4.4 สัญญาณเอาต์พุตจากวงจรขยายสัญญาณพัลส์ส่วนหน้า

4.3 การทดสอบวงจรมายสัญญาณพัลส์

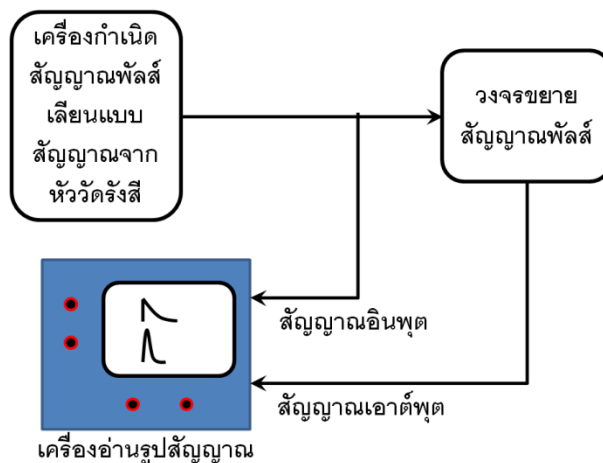
การทดสอบวงจรมายสัญญาณพัลส์ที่พัฒนาขึ้น ทำการทดสอบรูปสัญญาณเอาต์พุต และทดสอบความไม่เป็นเชิงเส้นของอัตราขยายสัญญาณพัลส์แบบอินทิกรัล (Integral Non-linearity)

4.3.1 เครื่องมือและอุปกรณ์

1. เครื่องกำเนิดสัญญาณพัลส์เลียนแบบสัญญาณจากหัววัดรังสี (Pulser) ของบริษัท Canberra รุ่น 814FP
2. เครื่องอ่านรูปสัญญาณ (Oscilloscope) ของบริษัท Tektronix รุ่น MSO3032
3. วงจรมายสัญญาณพัลส์ที่พัฒนาขึ้น

4.3.2 ขั้นตอนการทดสอบ

1. จัดอุปกรณ์ทดสอบดังแผนภาพภาพที่ 4.5



ภาพที่ 4.5 การจัดอุปกรณ์ทดสอบวงจรมายสัญญาณพัลส์

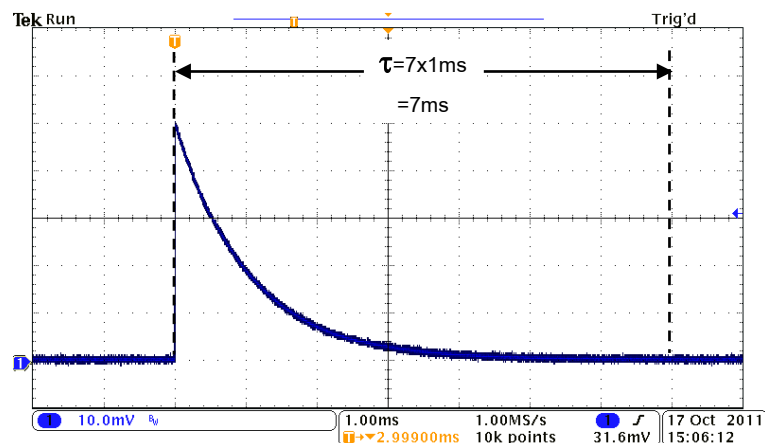
2. ทดสอบรูปสัญญาณพัลส์ของวงจรมายสัญญาณพัลส์ โดยป้อนสัญญาณพัลส์บวกขนาด 50 มิลลิโวลต์ จากเครื่องกำเนิดสัญญาณพัลส์เลียนแบบสัญญาณจากหัววัดรังสีให้กับวงจรมายสัญญาณพัลส์ที่พัฒนา ปรับอัตราขยายสัญญาณที่ 100 เท่า ทำการอ่านรูปสัญญาณพัลส์ด้านเอาต์พุตของวงจรมายสัญญาณพัลส์หลังจากปรับ Pole-Zero Cancellation ที่สมบูรณ์แล้วด้วยเครื่องอ่านรูปสัญญาณ

3. ทดสอบความไม่เป็นเชิงเส้นของอัตราขยายสัญญาณพัลส์แบบอินทิกรัล ปรับอัตราขยายสัญญาณที่ 100 เท่า และทำการปรับตั้งค่าความสูงของสัญญาณอินพุตจากเครื่อง

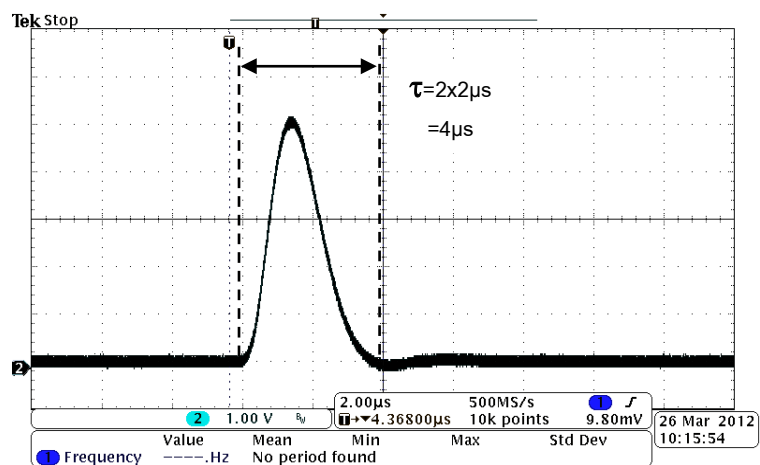
กำเนิดสัญญาณพัลส์เลียนแบบสัญญาณจากหัววัดรังสี บันทึกขนาดความสูงของสัญญาณเอาต์พุตในตารางที่ 4.2

4.3.3 ผลการทดสอบ

1. ผลการทดสอบรูปสัญญาณเอาต์พุตของวงจรขยายสัญญาณพัลส์ โดยการป้อนสัญญาณพัลส์บวก ขนาด 50 มิลลิโวลต์ จากเครื่องกำเนิดสัญญาณพัลส์เลียนแบบสัญญาณจากหัววัดรังสี ดังภาพที่ 4.6 ให้กับวงจรขยายสัญญาณพัลส์ที่พัฒนาขึ้น พบว่ารูปสัญญาณเอาต์พุตของวงจรหลังทำการปรับ Pole-zero Cancellation ดังภาพที่ 4.7 แสดงรูปสัญญาณพัลส์ที่เป็น Semi Gaussian มีค่าไรส์ไทม์ (Rise Time) 1 ไมโครวินาที และขนาดความกว้างของสัญญาณพัลส์ 4 ไมโครวินาที



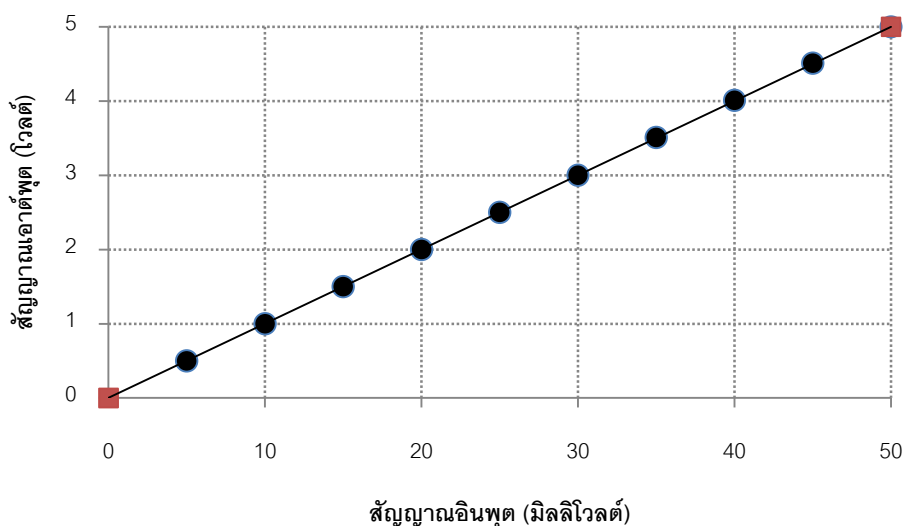
ภาพที่ 4.6 สัญญาณพัลส์จากเครื่องกำเนิดสัญญาณพัลส์เลียนแบบสัญญาณจากหัววัดรังสี



ภาพที่ 4.7 สัญญาณเอาต์พุตของวงจรขยายสัญญาณพัลส์ที่พัฒนาขึ้น

2. ผลการทดสอบความไม่เป็นเชิงเส้นของอัตราขยายสัญญาณพัลส์แบบอินทิกรัล ปรับอัตราขยายสัญญาณที่ 100 เท่า และเปลี่ยนขนาดความสูงของสัญญาณพัลส์ทางอินพุตตั้งแต่ 5, 10, 15, ..., 50 มิลลิโวลต์ ตามลำดับ บันทึกผลความสูงของสัญญาณพัลส์ทางด้านเอาต์พุตดังตารางที่ 4.2 และกราฟความสัมพันธ์ระหว่างสัญญาณอินพุตและสัญญาณเอาต์พุตดังภาพที่ 4.8 ตารางที่ 4.2 ขนาดความสูงของสัญญาณพัลส์ที่ได้จากวงจรรขยายสัญญาณพัลส์

สัญญาณพัลส์อินพุต (มิลลิโวลต์)	สัญญาณพัลส์เอาต์พุต (โวลต์)	อัตราขยาย
5	0.50	100
10	1.00	100
15	1.50	100
20	2.00	100
25	2.50	100
30	3.00	100
35	3.51	100.28
40	4.01	100.25
45	4.51	100.22
50	5.00	100



ภาพที่ 4.8 กราฟความสัมพันธ์ระหว่างสัญญาณอินพุตและสัญญาณเอาต์พุต

จากข้อมูลในตารางที่ 4.2 และภาพที่ 4.8 สามารถคำนวณหาค่าความไม่เป็นเชิงเส้นของ อัตรายายสัญญาณพัลส์แบบอินทิกรัล (INL) ได้จากสมการที่ 4.1

$$INL = \left| \frac{V-V_L}{V_{Max}} \right| \times 100\% \dots\dots\dots (4.1)$$

- โดยที่
- INL = ความไม่เป็นเชิงเส้นแบบอินทิกรัล
 - V = ความสูงของสัญญาณพัลส์ จากกราฟ ที่มีค่าเบี่ยงเบนสูงสุด
 - V_L = ความสูงของสัญญาณพัลส์ จากกราฟเส้นตรง
 - V_{Max} = ความสูงของสัญญาณเอาต์พุตสูงสุดที่ได้จากกราฟ

แทนค่าลงในสมการ 4.1 จะได้

$$INL = \left| \frac{3.51-3.50}{5} \right| \times 100\%$$

$$INL = 0.2\%$$

4.4 การทดสอบวงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดี่ยว

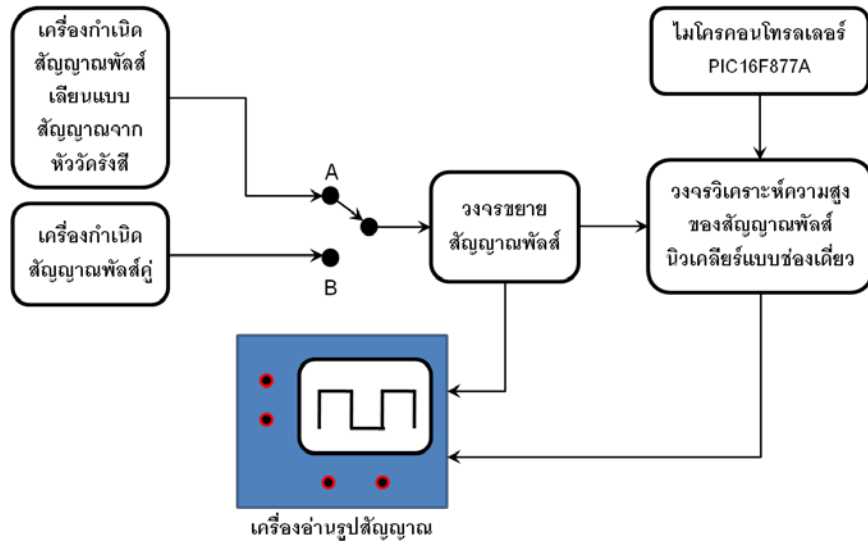
เป็นการทดสอบการทำงานของวงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดี่ยว ทำการทดสอบรูปสัญญาณเอาต์พุตและทดสอบความสามารถในการแยกสัญญาณพัลส์ที่เข้ามาใกล้ๆ กัน

4.4.1 เครื่องมือและอุปกรณ์

1. เครื่องกำเนิดสัญญาณพัลส์เลียนแบบสัญญาณจากหัววัดรังสี (Pulser) ของบริษัท Canberra รุ่น 814FP
2. เครื่องกำเนิดสัญญาณพัลส์คู่ (Pulse Pair Generator) ของ Canberra รุ่น 1407P
3. วงจรขยายสัญญาณพัลส์ที่พัฒนาขึ้น
4. เครื่องอ่านรูปสัญญาณ (Oscilloscope) ของบริษัท Tektronix รุ่น MSO3032
5. วงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดี่ยวที่พัฒนาขึ้น
6. วงจรนับจำนวนพัลส์พร้อมโปรแกรมตั้งเวลาโดยอาศัยการทำงานของ ไมโครคอนโทรลเลอร์ PIC16F877A

4.4.2 ขั้นตอนการทดสอบ

1. จัดอุปกรณ์ดังภาพที่ 4.9

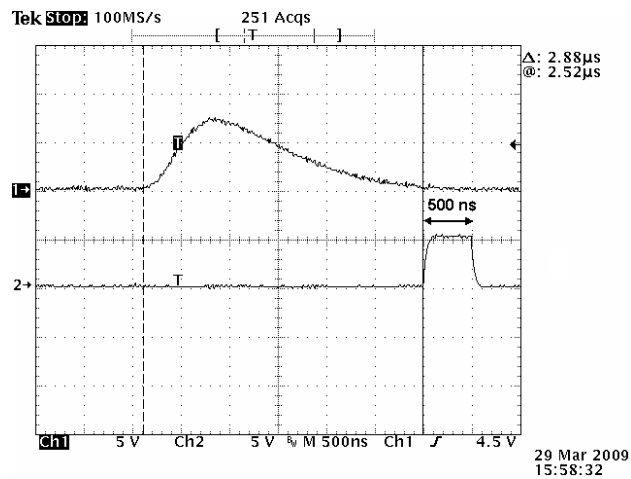


ภาพที่ 4.9 การจัดอุปกรณ์ทดสอบวงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดียว

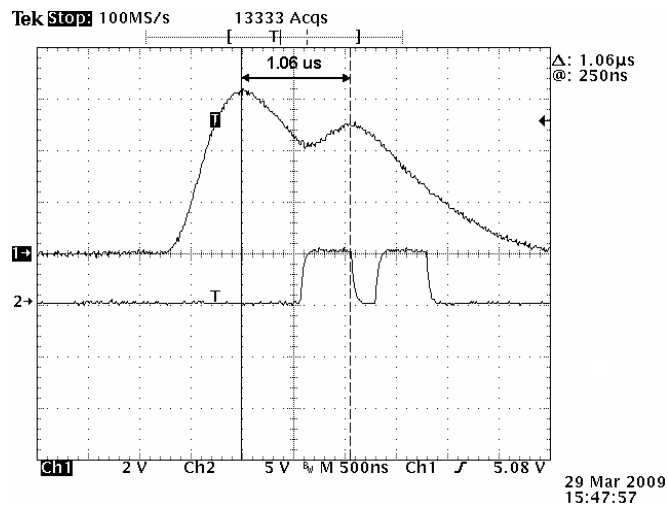
2. ทดสอบวงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดียว โดยให้จุดทดสอบอยู่ที่ตำแหน่ง A ระบบจะรับสัญญาณจากเครื่องกำเนิดสัญญาณพัลส์เลียนแบบสัญญาณจากหัววัดรังสี และขยายสัญญาณด้วยวงจรขยายสัญญาณพัลส์ที่พัฒนาขึ้นให้มีสัญญาณประมาณ 5 โวลต์ และป้อนสัญญาณพัลส์ให้กับวงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดียวที่ควบคุมการทำงานด้วยไมโครคอนโทรลเลอร์ โดยกำหนดให้วินโดว์มีค่าเท่ากับ 1 โวลต์ และดิสคริมิเนเตอร์ระดับล่างมีค่าเท่ากับ 4 โวลต์ และเมื่อปรับจุดทดสอบให้อยู่ที่ตำแหน่ง B ระบบจะรับสัญญาณจากเครื่องกำเนิดสัญญาณพัลส์คู่ ที่สามารถเปลี่ยนค่าความหน่วงเวลาของการเกิดสัญญาณพัลส์ที่ใกล้เคียงกันจากมากที่สุดไปหาค่าน้อยที่สุด ที่ยังเห็นสัญญาณลอจิกพัลส์ 2 พัลส์ปรากฏอยู่

4.4.3 ผลการทดสอบ

จากจุด A พบว่าสัญญาณเอาต์พุตที่ได้จากวงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดียว มีขนาด 500 ns ที่ขอบขาของของสัญญาณพัลส์นิวเคลียร์ตัดกับดิสคริมิเนเตอร์ระดับล่าง ดังภาพที่ 4.10 และจากจุด B พบว่าสัญญาณเอาต์พุตที่ได้จากวงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดียวสามารถแยกสัญญาณพัลส์ที่มีค่าความหน่วงเวลาที่ใกล้เคียงกันมากที่สุด ประมาณ 1.06 μ s ดังภาพที่ 4.11



ภาพที่ 4.10 สัญญาณเอาต์พุตของวงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดียว



ภาพที่ 4.11 สัญญาณเอาต์พุตเมื่อสัญญาณอินพุตเป็นพัลส์คู่ที่เกิดขึ้น

4.5 การทดสอบวงจรมับจำนวนพัลส์พร้อมโปรแกรมตั้งเวลา

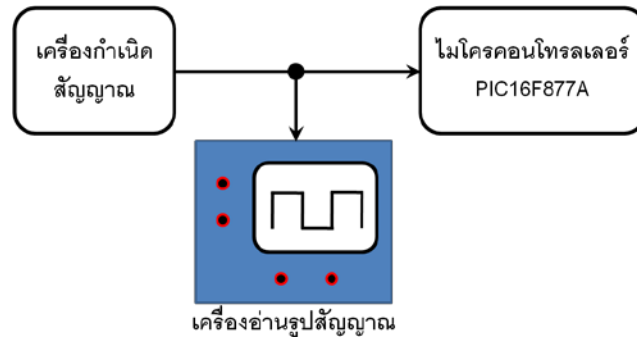
เป็นการทดสอบการทำงานของวงจรมับจำนวนพัลส์พร้อมโปรแกรมตั้งเวลาที่ได้พัฒนาขึ้นโดยอาศัยการทำงานของไมโครคอนโทรลเลอร์ PIC16F877A ทำการทดสอบตั้งเวลาและนับจำนวนสัญญาณพัลส์ลอจิกจากเครื่องกำเนิดสัญญาณที่มีความถี่ต่างๆ

4.5.1 เครื่องมือและอุปกรณ์

1. เครื่องกำเนิดสัญญาณ ของบริษัท GW Instek รุ่น GFG-3015
2. วงจรมับจำนวนพัลส์พร้อมโปรแกรมตั้งเวลาที่ได้พัฒนาขึ้น
3. เครื่องอ่านรูปสัญญาณ (Oscilloscope) ของบริษัท Tektronix รุ่น MSO3032

4.5.2 ขั้นตอนการทดสอบ

1. จัดอุปกรณ์ดังภาพที่ 4.12



ภาพที่ 4.12 การจัดอุปกรณ์ทดสอบวงจรนับจำนวนพัลส์พร้อมโปรแกรมตั้งเวลา

2. ทดสอบวงจรนับจำนวนพัลส์พร้อมโปรแกรมตั้งเวลาที่พัฒนาขึ้น โดยการป้อนสัญญาณพัลส์ลอจิกจากเครื่องกำเนิดสัญญาณที่มีความถี่ 10 เฮิรตซ์ ถึง 10 กิโลเฮิรตซ์ และบันทึกค่าจำนวนนับที่นับได้ดังตารางที่ 4.3 เพื่อหาค่าความผิดพลาดในการนับสัญญาณพัลส์ของวงจรถับจำนวนพัลส์พร้อมโปรแกรมตั้งเวลา

4.5.3 ผลการทดสอบ

วงจรถับจำนวนพัลส์พร้อมโปรแกรมตั้งเวลาที่พัฒนาสามารถนับจำนวนสัญญาณพัลส์ลอจิกจากเครื่องกำเนิดสัญญาณที่ความถี่ต่างๆ ดังตารางที่ 4.3

จากตารางที่ 4.3 สามารถคำนวณหาค่าความผิดพลาดในการนับจำนวนสัญญาณพัลส์ได้ดังสมการที่ 4.2

$$\% \text{ error} = \left| \frac{f_{in} - \text{Count}}{f_{in}} \right| \times 100\% \dots\dots\dots (4.2)$$

โดยที่ $\% \text{ error}$ = ค่าความผิดพลาด
 f_{in} = ความถี่อินพุต (Hz)
 Count = จำนวนนับที่นับได้จากระบบที่พัฒนาขึ้น (CPS)

แทนค่าลงในสมการ 4.2 จะได้

$$\% \text{ error} = \left| \frac{3,000 - 3,046}{3,000} \right| \times 100\%$$

$$\% \text{ error} = 1.5\%$$

ซึ่งจากสมการที่ 4.2 พบว่าวงจรนับจำนวนพัลส์พร้อมโปรแกรมตั้งเวลาที่พัฒนาขึ้นมีค่าความผิดพลาดสูงสุดในการนับจำนวนสัญญาณพัลส์เป็น 1.5%

ตารางที่ 4.3 จำนวนสัญญาณพัลส์ที่นับได้

ความถี่อินพุต (Hz)	จำนวนนับที่นับได้จาก โปรแกรมที่พัฒนาขึ้น (CPS)	ค่าความผิดพลาด (%)
10	10	0.0
30	30	0.0
50	50	0.0
70	70	0.0
90	90	0.0
100	101	1.0
300	304	1.3
500	507	1.4
700	710	1.4
900	911	1.2
1,000	1,010	1.0
3,000	3,046	1.5
5,000	5,074	1.5
7,000	7,099	1.4
9,000	9,137	1.5
10,000	10,148	1.5

4.6 การทดสอบความเป็นเชิงเส้นในการแปลงสัญญาณพัลส์

การทดสอบความเป็นเชิงเส้นในการแปลงสัญญาณพัลส์ ทดสอบการเปลี่ยนแปลงขนาดความสูงของสัญญาณพัลส์เป็นช่องพลังงาน โดยใช้โหมตสกีพในการทดสอบเพราะเป็นโหมตที่กำหนดค่าดิสคริเมเตอร์ระดับล่าง และวินโดว์ไว้คงที่ในโปรแกรมที่พัฒนาขึ้น

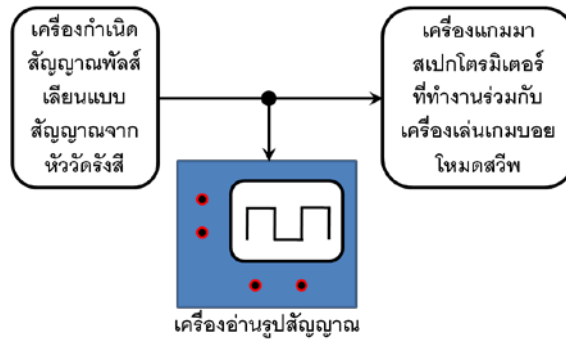
4.6.1 เครื่องมือและอุปกรณ์

1. เครื่องกำเนิดสัญญาณพัลส์เลียนแบบสัญญาณจากหัววัดรังสี (Pulser) ของบริษัท Canberra รุ่น 814FP
2. เครื่องอ่านรูปสัญญาณ (Oscilloscope) ของบริษัท Tektronix รุ่น MSO3032

3. เครื่องเกมมาสเปกโตรมิเตอร์ที่ทำงานร่วมกับเครื่องเล่นเกมบอย โหมดสวีฟ

4.6.2 ขั้นตอนการทดสอบ

1. จัดอุปกรณ์ดังภาพที่ 4.13



ภาพที่ 4.13 การจัดอุปกรณ์ทดสอบความเป็นเชิงเส้นในการแปลงสัญญาณพัลส์

2. ทดสอบความเป็นเชิงเส้นในการแปลงสัญญาณพัลส์ โดยปรับอัตราขยายสัญญาณของวงจรขยายสัญญาณพัลส์ที่ 140 เท่า และตั้งค่าสัญญาณพัลส์จากเครื่องกำเนิดสัญญาณพัลส์เลียนแบบสัญญาณจากหัววัดรังสีเป็น 5, 10, 15, ..., 35 มิลลิโวลต์ ตามลำดับ ป้อนให้กับเครื่องเกมมาสเปกโตรมิเตอร์ที่ทำงานร่วมกับเครื่องเล่นเกมบอย โหมดสวีฟ และบันทึกค่าช่องพลังงานที่ได้จากสเปกตรัมลงในตารางที่ 4.4

4.6.3 ผลการทดสอบ

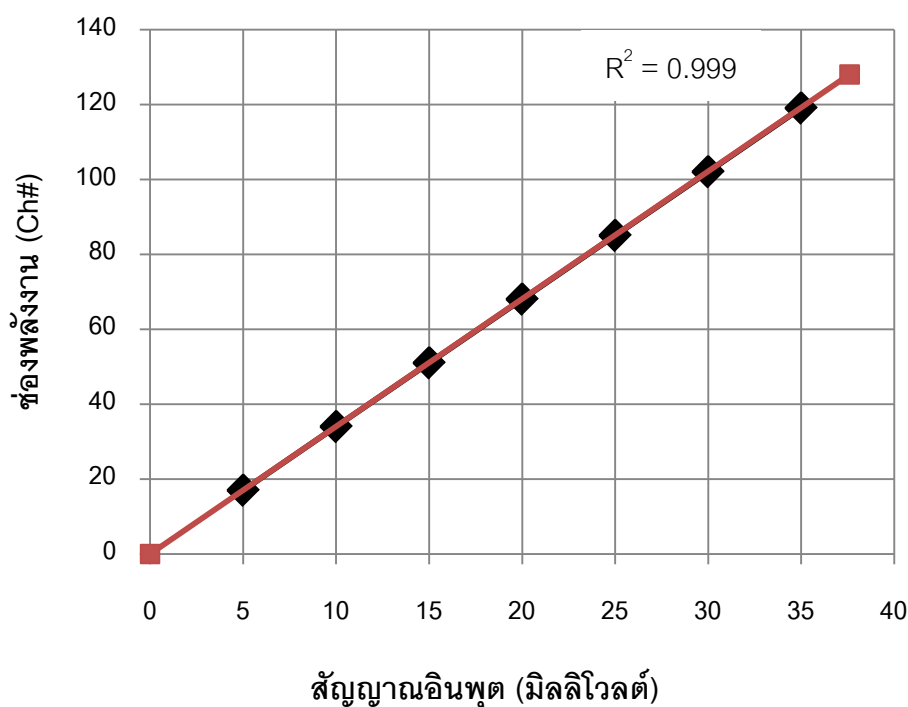
ผลการทดสอบความเป็นเชิงเส้นในการแปลงสัญญาณพัลส์ที่ได้จากการตั้งค่าอัตราขยายสัญญาณของวงจรขยายสัญญาณพัลส์ที่ 140 เท่า และเครื่องกำเนิดสัญญาณพัลส์เลียนแบบสัญญาณจากหัววัดรังสี แสดงดังตารางที่ 4.4 สามารถคำนวณหาความสัมพันธ์ระหว่างขนาดความสูงของสัญญาณพัลส์กับค่าช่องพลังงานได้ดังสมการที่ 4.3 และกราฟแสดงความเป็นเชิงเส้นของการแปลงสัญญาณพัลส์ดังภาพที่ 4.14

$$Ch\# = \frac{Ch_{Max}}{V_{Max}} \times V_{in} \times A_V \dots \dots \dots (4.3)$$

- โดยที่
- $Ch\#$ = ช่องพลังงาน
 - Ch_{Max} = ช่องพลังงานสูงสุด (128 ช่องพลังงาน)
 - V_{Max} = ขนาดความสูงของสัญญาณเอาต์พุตของวงจรขยายสัญญาณพัลส์ (5 โวลต์)
 - V_{in} = ขนาดความสูงของสัญญาณอินพุต
 - A_V = อัตราขยายสัญญาณ

ตารางที่ 4.4 ช่องพลังงานที่ได้จากการแปลงสัญญาณพัลส์ของระบบ

สัญญาณอินพุต (มิลลิโวลต์)	ช่องพลังงานที่วัดได้ (Ch#)	ช่องพลังงานที่ได้จาก การคำนวณ (Ch#)	ค่าความผิดพลาด (%)
5	17	17.28	1.65
10	35	35.57	1.60
15	54	53.85	-0.26
20	72	72.14	0.19
25	90	90.42	0.47
30	108	108.71	0.65
35	126	127.00	0.78



ภาพที่ 4.14 กราฟแสดงความสัมพันธ์เชิงเส้นของการแปลงสัญญาณพัลส์

4.7 การทดสอบเครื่องแกมมาสเปกโตรมิเตอร์ โหมดสำรวจรังสี

การทดสอบเครื่องแกมมาสเปกโตรมิเตอร์ โหมดสำรวจรังสี เป็นการทดสอบเพื่อดูความถูกต้อง และความน่าเชื่อถือของเครื่องแกมมาสเปกโตรมิเตอร์

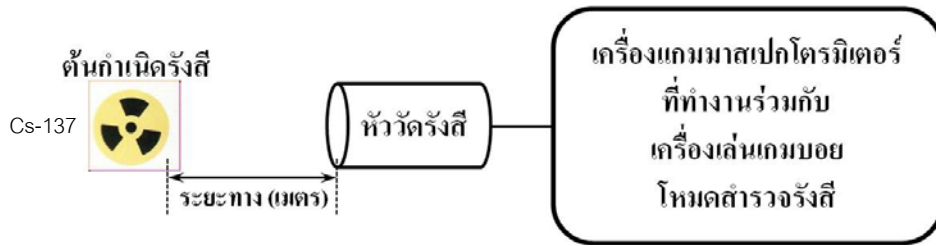
4.7.1 เครื่องมือและอุปกรณ์

1. ต้นกำเนิดรังสีซีซีเอ็ม 137

2. หัววัดรังสีชนิดซินทิลเลชันที่ทำงานร่วมกับฟิโนไฟโตไดโอด พร้อมทั้งวงจรรขยายสัญญาณ พัลส์ส่วนหน้า
3. เครื่องแกมมาสเปกโตรมิเตอร์ที่ทำงานร่วมกับเครื่องเล่นเกมบอย โหมดสำรวจรังสี

4.7.2 ขั้นตอนการทดสอบ

1. จัดอุปกรณ์ดังแผนภาพที่ 4.15



ภาพที่ 4.15 การจัดอุปกรณ์ทดสอบเครื่องแกมมาสเปกโตรมิเตอร์โหมดสำรวจรังสี

2. การทดสอบโหมดสำรวจรังสีด้วยต้นกำเนิดรังสีซีเซียม 137 ความแรงแรังสี 71.64 mCi อัตราปริมาณรังสี ที่ระยะ 1 เมตร เมื่อวันที่ 8 เดือนมกราคม พ.ศ. 2555 มีค่าเท่ากับ 24.42 มิลลิเรินท์เกน โดยจัดวางให้ระยะห่างจากจุดกึ่งกลางของหัววัดรังสีชนิดซินทิลเลชันที่ทำงานร่วมกับฟิโนไฟโตไดโอด ถึงจุดกึ่งกลางของต้นกำเนิดรังสีที่จุดปรับเทียบต่างๆ สามารถคำนวณจุดปรับเทียบได้จากสมการที่ 4.4 และบันทึกค่าปริมาณรังสีที่นับได้ดังตารางที่ 4.5

$$I_2 = I_1 \times \left(\frac{d_1}{d_2}\right)^2 \dots\dots\dots (4.4)$$

- โดยที่
- I_2 = ปริมาณรังสี ณ จุดที่ต้องการปรับเทียบ
 - I_1 = ปริมาณรังสี ณ ที่จุด 1 เมตร
 - d_1 = ระยะทาง 1 เมตร
 - d_2 = ระยะทาง ณ จุดที่ต้องการปรับเทียบ

ตารางที่ 4.5 ปริมาณรังสีที่นับวัดได้ ณ จุดปรับเทียบต่างๆ ของเครื่องสำรวจรังสีที่พัฒนาขึ้น

ระยะทาง ณ จุด ปรับเทียบ (เมตร)	ปริมาณรังสีที่คำนวณได้ ณ จุดปรับเทียบต่างๆ (mR/h)	ปริมาณรังสีที่นับวัดได้จาก เครื่องสำรวจรังสีที่พัฒนา (mR/h)	ค่าความผิดพลาด (%)
ย่านวัด x1			
0.52	87.43	90.88	3.9
0.60	65.67	66.78	1.6
0.76	40.93	41.63	1.7
1.08	20.27	21.32	5.1
ย่านวัด x0.1			
1.70	8.18	8.34	1.9
1.90	6.55	6.46	1.3
2.40	4.10	4.26	3.9
3.40	2.04	2.09	2.4
ย่านวัด x0.01			
5.40	0.81	0.79	2.4
6.20	0.61	0.62	1.6
7.40	0.43	0.42	2.3
10.60	0.21	0.20	4.7

4.7.3 ผลการทดสอบ

จากตารางที่ 4.5 เห็นได้ว่าเครื่องแกมมาสเปกโตรมิเตอร์ที่ทำงานร่วมกับเครื่องเล่นเกมบอย ในโหมดสำรวจรังสี มีค่าความผิดพลาดสูงสุดในการวัดปริมาณรังสีของแต่ละย่านวัด เท่ากับ 5.1, 3.9 และ 4.7 เปอร์เซ็นต์ ตามลำดับ

4.8 การทดสอบเครื่องแกมมาสเปกโตรมิเตอร์ โหมดสแกน

การทดสอบเครื่องแกมมาสเปกโตรมิเตอร์ โหมดสแกน เป็นการทดสอบสแกนสเปกตรัมพลังงานของรังสี จากต้นกำเนิดรังสีซีเซียม 137 และโคบอลต์ 60 เพื่อคำนวณหาค่าความสามารถในการแจกแจงพลังงานของเครื่องแกมมาสเปกโตรมิเตอร์

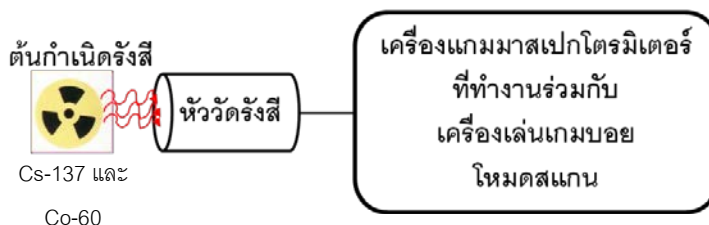
4.8.1 เครื่องมือและอุปกรณ์

1. ต้นกำเนิดรังสีซีเซียม 137 และโคบอลต์ 60 (Co-60 : พลังงาน 1.17 และ 1.33 MeV)
2. หัววัดรังสีชนิดซินทิลเลชันที่ทำงานร่วมกับพินโฟโตไดโอด พร้อมทั้งวงจรมหาสัญญาณพัลส์ส่วนหน้า

3. เครื่องแกมมาสเปกโตรมิเตอร์ที่ทำงานร่วมกับเครื่องเล่นเกมบอย โหมดสแกน

4.8.2 ขั้นตอนการทดสอบ

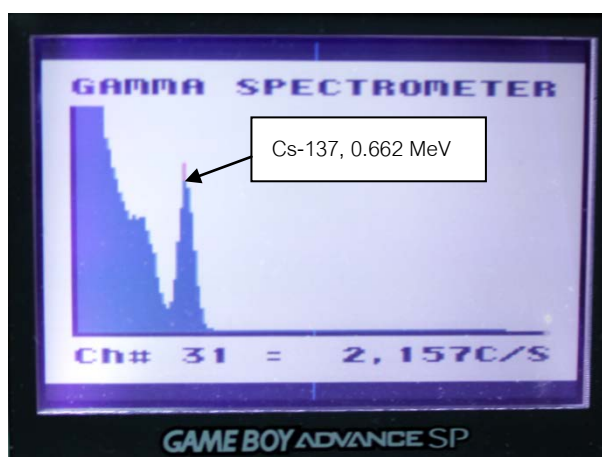
1. จัดอุปกรณ์ดังแผนภาพที่ 4.16



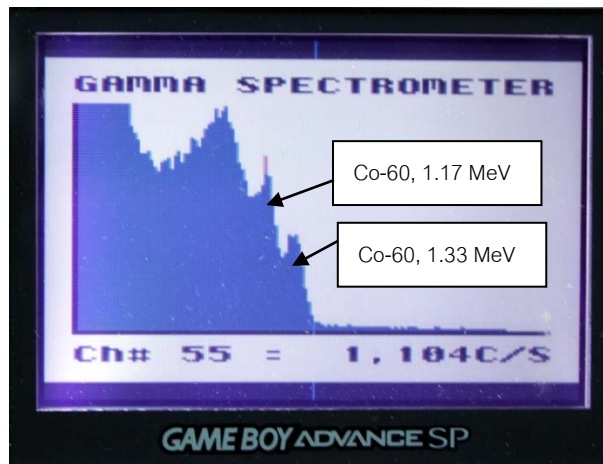
ภาพที่ 4.16 การจัดอุปกรณ์ทดสอบเครื่องแกมมาสเปกโตรมิเตอร์ในโหมดสแกน

2. เลือกโหมดการทำงานเป็นโหมดสแกน กำหนดค่าดิสคริมีเนเตอร์ระดับต่ำเท่ากับ 40 มิลลิโวลต์ วินโดว์เท่ากับ 40 มิลลิโวลต์ และค่าเวลาในการสแกนแต่ละช่องสัญญาณเท่ากับ 5 วินาที อัตราขยายสัญญาณที่ 140 เท่า นำต้นกำเนิดรังสีซีซีเซียม 137 มาวางให้กึ่งกลางของหัววัดรังสีอยู่ระหว่างกึ่งกลางของต้นกำเนิดรังสี และเริ่มวัดรังสีพร้อมทั้งแสดงสเปกตรัมพลังงานรังสีดังภาพที่ 4.17 และทำการเปลี่ยนต้นกำเนิดรังสีจากซีซีเซียม 137 เป็นโคบอลต์ 60 และเริ่มวัดรังสีพร้อมทั้งแสดงสเปกตรัมพลังงานรังสีดังภาพที่ 4.18 และนำต้นกำเนิดรังสีซีซีเซียม 137 และโคบอลต์ 60 มาวางให้อยู่กึ่งกลางของหัววัดรังสีอยู่ระหว่างกึ่งกลางของต้นกำเนิดรังสี และเริ่มวัดรังสีพร้อมทั้งแสดงสเปกตรัมพลังงานรังสีดังภาพที่ 4.19

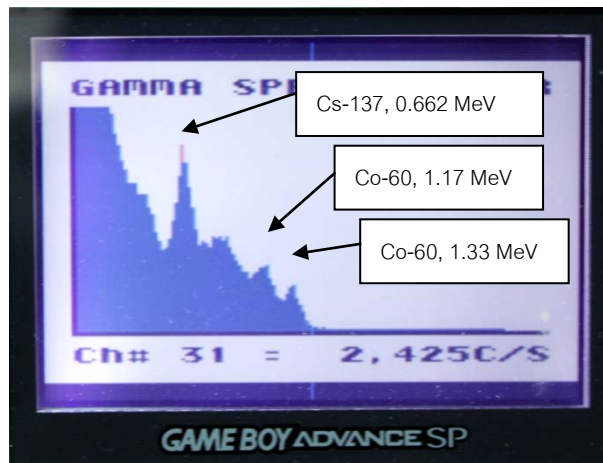
4.8.3 ผลการทดสอบ



ภาพที่ 4.17 สเปกตรัมพลังงานรังสีโหมดสแกน จากต้นกำเนิดรังสีซีซีเซียม 137



ภาพที่ 4.18 สเปกตรัมพลังงานรังสีโหมดสแกน จากต้นกำเนิดรังสีโคบอลต์ 60



ภาพที่ 4.19 สเปกตรัมพลังงานรังสีโหมดสแกน จากต้นกำเนิดรังสีซีเซียม 137 และโคบอลต์ 60

จากภาพที่ 4.17 สามารถคำนวณหาค่า FWHM ได้ดังสมการที่ 4.5

$$FWHM = \frac{ch_r - ch_l}{ch_c} \times 100\% \dots \dots \dots (4.5)$$

โดยที่ $FWHM$ = ความสามารถในการแจกแจงพลังงาน

ch_r = ช่องพลังงานที่มีค่าจำนวนนับครึ่งหนึ่งของพีคทางด้านขวา

ch_l = ช่องพลังงานที่มีค่าจำนวนนับครึ่งหนึ่งของพีคทางด้านซ้าย

ch_c = ช่องพลังงาน ณ จุดพีค

แทนค่าลงในสมการ 4.5 จะได้

$$FWHM = \frac{33 - 29}{31} \times 100\% \\ = 12.9\%$$

4.9 การทดสอบเครื่องแกมมาสเปกโตรมิเตอร์ โหมดสวิต

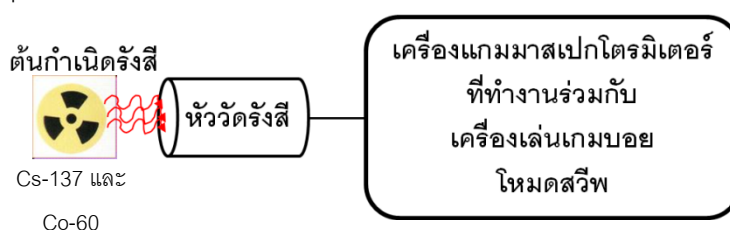
การทดสอบเครื่องแกมมาสเปกโตรมิเตอร์ โหมดสวิต เป็นการทดสอบเพื่อนำค่าของพลังงานที่ได้จากการสแกนสเปกตรัมพลังงานของรังสีมาสร้างกราฟเปรียบเทียบพลังงาน (Energy Calibration Curve)

4.9.1 เครื่องมือและอุปกรณ์

1. ต้นกำเนิดรังสีซีเซียม 137 และโคบอลต์ 60
2. หัววัดรังสีชนิดซินทิลเลชันที่ทำงานร่วมกับฟิโนไดโอด พร้อมทั้งวงจรรขยายสัญญาณพัลส์ส่วนหน้า
3. เครื่องแกมมาสเปกโตรมิเตอร์ที่ทำงานร่วมกับเครื่องเล่นเกมบอย โหมดสวิต

4.9.2 ขั้นตอนการทดสอบ

1. จัดอุปกรณ์ดังแผนภาพที่ 4.20

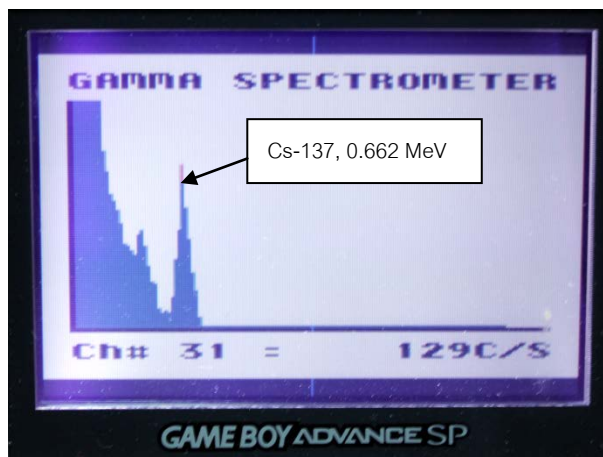


ภาพที่ 4.20 การจัดอุปกรณ์ทดสอบเครื่องแกมมาสเปกโตรมิเตอร์ในโหมดสวิต

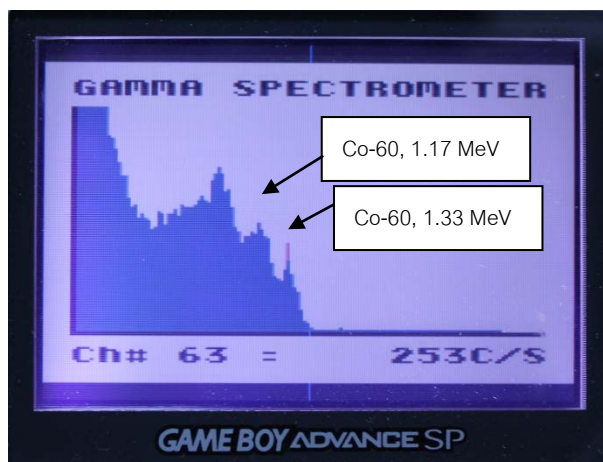
2. เลือกโหมดการทำงานเป็นโหมดสวิต ในโหมดนี้ได้ออกแบบและพัฒนาโปรแกรมให้กำหนดค่าต่างๆ ของ SCA ไว้อัตโนมัติ คือ ดิสคริเมเนเตอร์ระดับต่ำเท่ากับ 0 โวลต์ วินโดว์เท่ากับ 0.04 โวลต์ และเวลาในการสแกนแต่ละช่องพลังงานเท่ากับ 0.1 วินาที ถ้าต้องการเพิ่มจำนวนเวลาในการสแกนสามารถเพิ่มได้โดยการกำหนดค่าจำนวนรอบในการสวิต ซึ่งออกแบบให้สามารถกำหนดรอบในการสวิตได้สูงสุด 256 รอบ ทำการทดสอบโหมดสวิตโดยปรับอัตราขยายสัญญาณที่ 140 เท่า ตั้งจำนวนรอบในการสวิตเป็น 6 รอบ นำต้นกำเนิดรังสีซีเซียม 137 มาวางให้กึ่งกลางของหัววัดรังสีอยู่ระหว่างกึ่งกลางของต้นกำเนิดรังสี และเริ่มวัดรังสีพร้อมทั้งแสดงสเปกตรัมพลังงานรังสีดังภาพที่ 4.21 และทำการเปลี่ยนต้นกำเนิดรังสีจากซีเซียม 137 เป็นโคบอลต์ 60 และเริ่มวัดรังสีพร้อมทั้งแสดงสเปกตรัมพลังงานรังสีดังภาพที่ 4.22 และนำต้นกำเนิดรังสีซีเซียม 137

และโคบอลต์ 60 มาวางให้อยู่กึ่งกลางของหัววัดรังสีอยู่ระหว่างกึ่งกลางของต้นกำเนิดรังสี และเริ่มวัดรังสีพร้อมทั้งแสดงสเปกตรัมพลังงานรังสีดังภาพที่ 4.23

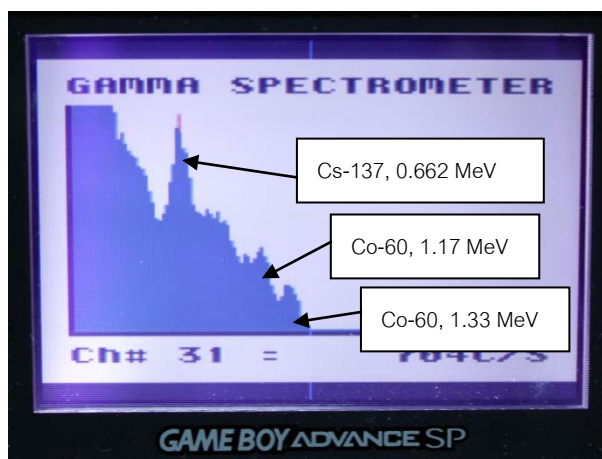
4.9.3 ผลการทดสอบ



ภาพที่ 4.21 สเปกตรัมพลังงานรังสีโหมดสวீฟ จากต้นกำเนิดรังสีซีเซียม 137

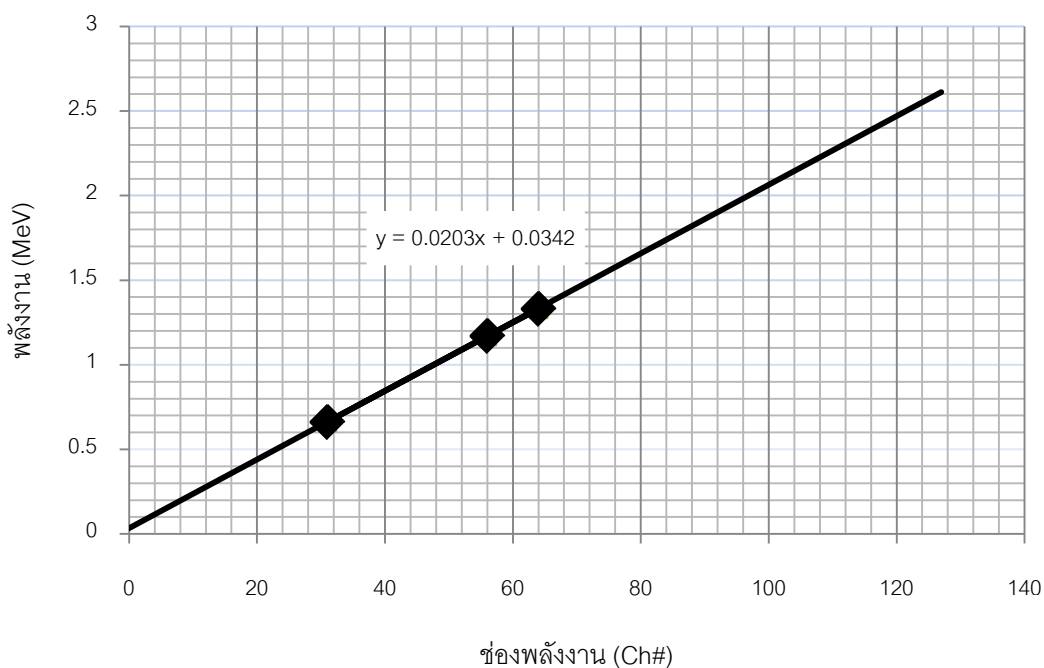


ภาพที่ 4.22 สเปกตรัมพลังงานรังสีโหมดสวี่ฟ จากต้นกำเนิดรังสีโคบอลต์ 60



ภาพที่ 4.23 สเปกตรัมพลังงานรังสีโหมดสวี่ฟ จากต้นกำเนิดรังสีซีเซียม 137 และโคบอลต์ 60

จากภาพที่ 4.23 เมื่อเลื่อนเคอร์เซอร์มาที่พีคพลังงาน 0.662 MeV เกิดที่ช่องพลังงาน 31 พีคพลังงาน 1.17 MeV เกิดที่ช่องพลังงาน 55 พีคพลังงาน 1.33 MeV เกิดที่ช่องพลังงาน 63 สามารถนำไปสร้างกราฟเปรียบเทียบพลังงาน ดังภาพที่ 4.24



ภาพที่ 4.24 กราฟเปรียบเทียบพลังงานจากการวัดสเปกตรัมด้วยหัววัด CsI(Tl)

บทที่ 5

สรุปผลการวิจัย และข้อเสนอแนะ

5.1 สรุปผลการวิจัย

สรุปผลการทดสอบเครื่องแกมมาสเปกโตรมิเตอร์ที่ทำงานร่วมกับเครื่องเล่นเกมบอย

5.1.1 ผลการทดสอบแหล่งจ่ายไฟฟ้าคิกดาต้า

วงจรแหล่งจ่ายไฟฟ้าคิกดาต้าขนาด ± 12 โวลต์ ที่ใช้เป็นแหล่งจ่ายกำลังไฟฟ้าให้กับหัววัดรังสี วงจรขยายสัญญาณพัลส์ส่วนหน้า วงจรขยายสัญญาณพัลส์ และแรงดันไฟฟ้าอ้างอิงของวงจร SCA พบว่าแหล่งจ่ายไฟฟ้าคิกดาต้าแต่ละชุดมีค่าสัญญาณรบกวนและริบเบิลในขณะจ่ายกำลังไฟฟ้าให้กับระบบประมาณ 5 มิลลิโวลต์ และมีอัตราสิ้นเปลืองกำลังไฟฟ้าเท่ากับ 1.38 วัตต์ สามารถใช้งานได้ต่อเนื่อง 11 ชั่วโมง

5.1.2 ผลการทดสอบวงจรขยายสัญญาณพัลส์ส่วนหน้า

วงจรขยายสัญญาณพัลส์ส่วนหน้ามีสัญญาณพัลส์ขนาด 9 มิลลิโวลต์ จากการวัดต้นกำเนิดรังสีซีเซียม 137 พลังงาน 662 กิโลอิเล็กตรอนโวลต์ (KeV) และมีค่าเวลาการสลายตัว (Decay time) ประมาณ 50 ไมโครวินาที จากการทดสอบพบว่าความสัมพันธ์ระหว่างสัญญาณพัลส์ที่ได้จากวงจรขยายสัญญาณพัลส์ส่วนหน้ากับพลังงานจากต้นกำเนิดรังสีมีอัตราขยายสัญญาณประมาณ 12 mV/MeV

5.1.3 ผลการทดสอบวงจรขยายสัญญาณพัลส์

วงจรขยายสัญญาณพัลส์ที่พัฒนาขึ้นมีอัตราขยายสัญญาณสูงสุด 200 เท่า สามารถสร้างสัญญาณพัลส์แบบ Semi Gaussian ที่มีค่าไรส์ไทม์ 1 ไมโครวินาที และความกว้างของสัญญาณพัลส์ 4 ไมโครวินาที มีค่าความไม่เป็นเชิงเส้นของอัตราขยายสัญญาณแบบอินทิกรัล 0.2 เปอร์เซ็นต์ และสามารถขยายสัญญาณให้มีความสูงของสัญญาณตั้งแต่ 0 – 5 โวลต์

5.1.4 ผลการทดสอบวงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดียว

สัญญาณเอาต์พุตที่ได้จากวงจรวิเคราะห์ความสูงของสัญญาณพัลส์นิวเคลียร์แบบช่องเดียว มีขนาดความกว้างของสัญญาณพัลส์ 500 ns ที่ขอบขาของสัญญาณพัลส์นิวเคลียร์ที่ตัดกับดิสคริเมเนเตอร์ระดับล่าง และสามารถแยกสัญญาณพัลส์ที่มีค่าความหน่วงเวลาที่ใกล้เคียงกันมากที่สุด ประมาณ 1.06 μ s

5.1.5 ผลการทดสอบวงจรตั้งเวลาและวงจรนับจำนวนพัลส์

วงจรตั้งเวลาและวงจรนับจำนวนพัลส์มีค่าความผิดพลาดในการนับจำนวนสัญญาณพัลส์เป็น 1.5 เปอร์เซ็นต์ เนื่องจากปัญหาต่างๆ คือเครื่องกำเนิดสัญญาณ เครื่องอ่านรูปสัญญาณ และการพัฒนาโปรแกรมสำหรับนับจำนวนสัญญาณพัลส์ต่อหน่วยของเวลาที่อาจเกิดการคลาดเคลื่อนของเวลา

5.1.6 ผลการทดสอบความเป็นเชิงเส้นในการแปลงสัญญาณพัลส์

ผลการทดสอบความเป็นเชิงเส้นในการแปลงสัญญาณพัลส์พบว่ามีความผิดพลาดสูงสุดที่ 1.65 เปอร์เซ็นต์ ซึ่งเกิดจากผลต่างของค่าที่คำนวณเป็นจุดทศนิยมแต่ค่าที่วัดได้เป็นเลขจำนวนเต็มและพบว่ามีค่าความเป็นเชิงเส้น $R^2 = 0.999$ ซึ่งมีความน่าเชื่อถือในการแปลงสัญญาณพัลส์

5.1.7 ผลการทดสอบเครื่องแกมมาสเปกโตรมิเตอร์ โหมดสำรวจรังสี

ผลการทดสอบเครื่องแกมมาสเปกโตรมิเตอร์ โหมดสำรวจรังสี ในแต่ละย่านวัดพบว่ามีความผิดพลาดสูงสุดในการวัดปริมาณรังสีของย่านวัด $x1$, $x0.1$ และ $x0.01$ mR/h เท่ากับ 5.1, 3.9 และ 4.7 เปอร์เซ็นต์ ตามลำดับ

5.1.8 ผลการทดสอบเครื่องแกมมาสเปกโตรมิเตอร์ โหมดสแกน

ผลการทดสอบเครื่องแกมมาสเปกโตรมิเตอร์ โหมดสแกน พบว่าสามารถกำหนดค่าดิสคริมีเนเตอร์ระดับล่างและค่าของวินโดว์ได้ตามที่ต้องการ และเป็นสัดส่วนต่อค่าพลังงาน กล่าวคือสามารถเลือกวัดเฉพาะช่วงพลังงานที่ต้องการได้ และมีความสามารถในการแจกแจงพลังงานเป็น 12.9 เปอร์เซ็นต์

5.1.9 ผลการทดสอบเครื่องแกมมาสเปกโตรมิเตอร์ โหมดสวิตช์

ผลการทดสอบเครื่องแกมมาสเปกโตรมิเตอร์ โหมดสวิตช์ พบว่าสามารถกำหนดจำนวนรอบในการสวิตช์ ตั้งแต่ 1 – 255 รอบ โดยใช้เวลาในการสวิตช์ในแต่ละรอบประมาณ 13 วินาที และสามารถวัดและแสดงผลของสเปกตรัมได้เป็นที่น่าพอใจ จากกราฟเปรียบเทียบพลังงานจะได้สมการ $y = 0.0203x + 0.0342$ เพื่อเปรียบเทียบค่าพลังงานที่ได้จากการวัดสเปกตรัมของต้นกำเนิดรังสีที่ไม่ทราบค่าพลังงาน

5.2 ปัญหาและอุปสรรค

5.2.1 เนื่องจากในงานวิจัยนี้ต้องการพัฒนาเครื่องมือให้มีขนาดเล็ก สามารถเคลื่อนย้ายได้สะดวกจึงจำเป็นต้องสร้างแหล่งจ่ายไฟฟ้าศักดาต่ำแบบสวิตชิงมาใช้งานแทนแหล่งจ่ายไฟฟ้าศักดาต่ำแบบลิเนียร์ เป็นเหตุให้มีสัญญาณรบกวนที่หัววัดรังสีและวงจรรขยายสัญญาณพัลส์ทำให้ประสิทธิภาพในการวัดรังสีแกมมาลดลง

5.2.2 เนื่องจากในงานวิจัยนี้เลือกใช้หัววัดรังสีชนิดผลึกซินทิลเลเตอร์ CsI(Tl) ขนาด 1x1 เซนติเมตร ซึ่งมีขนาดเล็ก ทำให้ปริมาณการตกกระทบของรังสีมีพื้นที่น้อยลง มีผลให้ประสิทธิภาพในการวัดรังสีลดลง

5.2.3 ในการพัฒนาโปรแกรมยังไม่สามารถเรียกใช้หน่วยความจำได้อย่างเต็มรูปแบบ เนื่องจากเทคโนโลยีและโปรแกรมที่ใช้ได้หยุดการพัฒนา จึงทำให้การนำมาใช้มีข้อมูลในการศึกษาและพัฒนาจำกัด

5.3 ข้อเสนอแนะ

5.3.1 หัววัดรังสีที่ใช้สำหรับงานวิจัยนี้เป็นหัววัดรังสีชนิดผลึกซินทิลเลเตอร์ CsI(Tl) ขนาด 10 x 10 มิลลิเมตร ที่ทำงานร่วมกับฟิโนโฟโตไดโอด ซึ่งเป็นหัววัดรังสีที่มีขนาดเล็กที่สามารถวัดรังสีแบบแจกแจงพลังงานได้ แต่จากงานวิจัยนี้พบว่าประสิทธิภาพของหัววัดรังสีชนิดนี้ยังไม่ดีพอ เนื่องจากปริมาณการตกกระทบของรังสีต่อพื้นที่มีค่าน้อย ซึ่งถ้าใช้ผลึกซินทิลเลเตอร์ที่มีขนาดใหญ่ขึ้นก็จะทำให้มีประสิทธิภาพในการวัดรังสีดีขึ้น

5.3.2 ปัจจุบันเทคโนโลยีของเครื่องเล่นแบบพกพาได้มีการพัฒนาโดยการใส่ไมโครโพรเซสเซอร์ที่มีความเร็วในการประมวลผลมากขึ้นประกอบกับการพัฒนาโปรแกรมเพื่อติดต่อกับเครื่องเล่นแบบพกพาทำได้ง่ายขึ้นและสามารถติดต่อกับหน่วยความจำภายนอกได้อย่างอิสระ ซึ่งสามารถนำมาใช้แทนเครื่องเล่นบอยรูนแอ็ดวานซ์ที่ใช้อยู่ในงานวิจัย

รายการอ้างอิง

- [1] สิริยาภรณ์ แสงอรุณ. การพัฒนาระบบมอนิเตอร์ริงส์เกมมาชนิดเคลื่อนย้ายได้ที่เชื่อมโยงด้วย จีพีเอส. วิทยานิพนธ์ปริญญาโทมหาบัณฑิต, ภาควิชาวิศวกรรมเทคโนโลยี คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2547.
- [2] Steve Willis. Gameboy Digital Sampling Oscilloscope (GBDSO). Elektor electronics magazine October and November 2000: 34-39 and 12-15.
- [3] Marcel Cremmel. Gameboy Electro-cardiogram (GBECG). Elektor electronics magazine October 2006: 32-41.
- [4] อรรถพร ภัทรสุมันต์. เอกสารประกอบการสอนการวัดรังสี. ภาควิชาวิศวกรรมเทคโนโลยี คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2545.
- [5] สุวิทย์ ปุณณชัยยะ. NUCLEAR RADIATION DETECTION AND INSTRUMENTATION. เอกสารประกอบการสอนวิชา 2111606 ภาควิชาวิศวกรรมเทคโนโลยี คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2538.
- [6] ชูชัย ธนาสารตั้งเจริญและคณะ. การใช้งาน Z80. กรุงเทพฯ: ฟิสิกส์เซ็นเตอร์, 2540.
- [7] อีรวัดน์ ประกอบผล. คู่มือการเขียนโปรแกรม ภาษา C. กรุงเทพฯ: ชิมพลีฟาย, 2553.
- [8] อรพิน ประวัตินิรุทธิ์. คู่มือเรียนภาษาซีฉบับปรับปรุงใหม่. กรุงเทพฯ: โปรวิชั่น, 2552.
- [9] โกศล วงกานต์. การพัฒนาเครื่องนับรังสีชนิดเคลื่อนย้ายได้แบบประหยัดโดยใช้ CPLD. วิทยานิพนธ์ปริญญาโทมหาบัณฑิต, ภาควิชาวิศวกรรมเทคโนโลยี คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2552.

ภาคผนวก

ภาคผนวก ก

คู่มือการใช้งาน

เครื่องแกมมาสเปกโตรมิเตอร์ที่ทำงานร่วมกับเครื่องเล่นเกมบอย

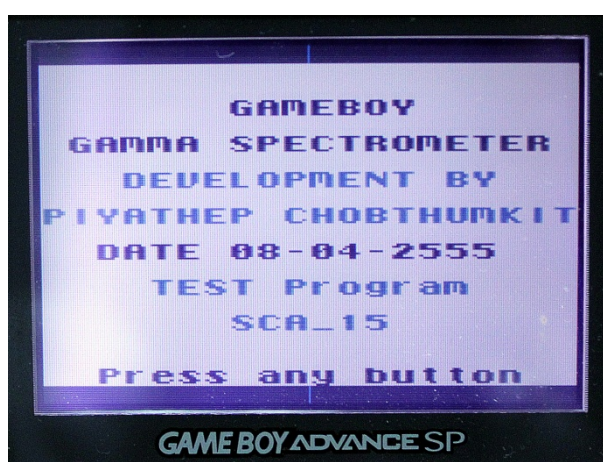
คู่มือการใช้งานเครื่องแกมมาสเปกโตรมิเตอร์ที่ทำงานร่วมกับเครื่องเล่นเกมบอย

1. เชื่อมต่ออุปกรณ์ดังภาพที่ 1 และเปิดเครื่องแกมมาสเปกโตรมิเตอร์ที่ทำงานร่วมกับเครื่องเล่นเกมบอย



ภาพที่ 1

2. เมื่อเปิดเครื่องจะปรากฏหน้าจอ ดังภาพที่ 2 แสดงข้อมูลต่างๆ ที่เกี่ยวกับผู้พัฒนาโปรแกรม และแสดงคำว่า "Press any button" เมื่อกดปุ่มใดๆ จะเข้าสู่หน้าจอสำหรับเลือกโหมดการทำงาน ดังภาพที่ 3



ภาพที่ 2

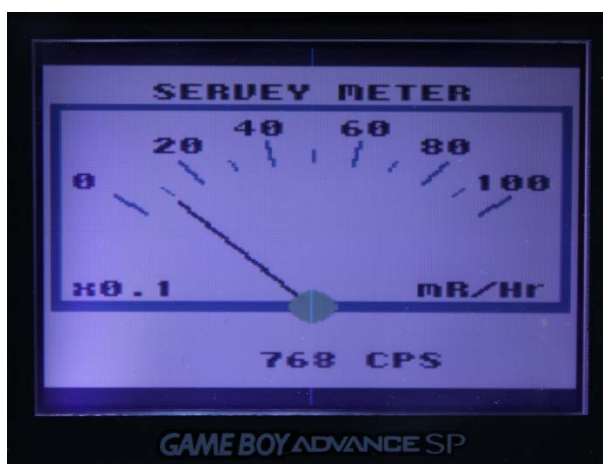


ภาพที่ 3

3. เมื่อเข้าสู่หน้าจอสำหรับเลือกโหมดการทำงานแสดงดังภาพที่ 3 จะปรากฏโหมดการทำงาน 2 โหมดคือ 1) Survey Meter และ 2) Gamma spec โดยกดปุ่ม UP หรือปุ่ม DOWN เพื่อเลือกโหมดการทำงาน จากนั้นกดปุ่ม Select เพื่อเข้าโหมดการทำงานที่เลือกไว้
4. โหมด Survey Meter เมื่อเข้ามาในโหมดนี้ จะปรากฏหน้าจอดังภาพที่ 4 ในโหมดนี้จะมีการแสดงผล 2 รูปแบบคือ 1) หน่วยการวัดอัตราการรับปริมาณรังสี (Exposure Rate) ในย่าน 0-100 mR/hr และ 2) หน่วยการวัดอัตรานับรังสี (Count Rate) ในย่าน $0-2.55 \times 10^4$ CPS ซึ่งสามารถเลือกหน่วยได้โดยการกดปุ่ม UP เพื่อเลือกหน่วยการวัดอัตราการรับปริมาณรังสีดังภาพที่ 4 และกดปุ่ม DOWN เพื่อเลือกหน่วยการวัดอัตรานับรังสีดังภาพที่ 5 และหากต้องการกลับไปยังหน้าจอสำหรับเลือกโหมดการทำงานให้กดปุ่ม B



ภาพที่ 4



ภาพที่ 5

5. โหมด Gamma spec (Gamma spectrometer) ดังภาพที่ 6 เมื่อเข้ามาในโหมดนี้ จะปรากฏหน้าจอดังภาพที่ 7 ประกอบด้วย 2 โหมดย่อยคือ 1) โหมด Scan spectrometer และ 2) โหมด Sweep spectrometer ซึ่งสามารถเลือกโหมดได้โดยการกดปุ่ม UP สำหรับเลือกโหมด Scan spectrometer ดังภาพที่ 7 และกดปุ่ม DOWN เพื่อเลือกโหมด Sweep spectrometer ดังภาพที่ 8 จากนั้นให้กดปุ่ม Select เพื่อเข้าโหมดการทำงานที่เลือกไว้



ภาพที่ 6



ภาพที่ 7



ภาพที่ 8

6. โหมด Scan spectrometer เป็นโหมดสำหรับการสแกนสเปกตรัมพลังงานรังสี ที่สามารถกำหนดค่า LLD, Windows (ΔE) และเวลาในการสแกนแต่ละช่องสัญญาณ ดังภาพที่ 9, 10, 11 ตามลำดับ ซึ่งการส่งค่าต่างๆ นั้นจะต้องกดปุ่ม Select เพื่อส่งค่าที่กำหนดทุกๆ ครั้ง



ภาพที่ 9

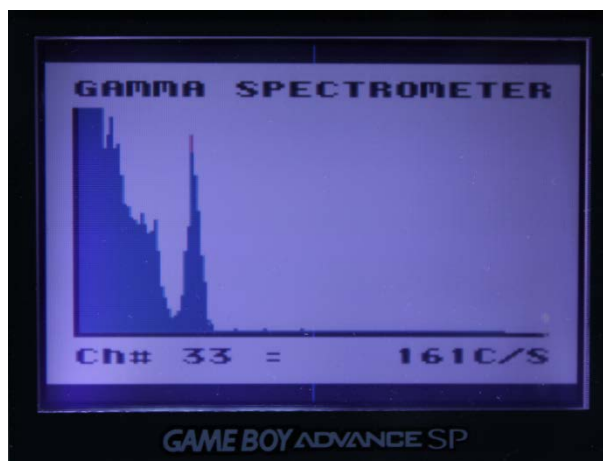


ภาพที่ 10



ภาพที่ 11

7. เมื่อกำหนดค่าต่างๆ และกดปุ่ม Select เพื่อส่งค่าเวลาแล้ว จะปรากฏหน้าจอแสดงสเปกตรัมพลังงานรังสีที่ทำการวัด ดังภาพที่ 12 และสามารถเลื่อนเคอร์เซอร์โดยการกดปุ่ม LEFT หรือปุ่ม RIGHT เพื่ออ่านค่าจำนวนนับต่อเวลาที่ได้ในแต่ละช่องพลังงาน (Ch#) และสามารถขยายและลดหน้าจอในแนวสูง โดยการกดปุ่ม UP หรือปุ่ม DOWN เมื่อวิเคราะห์สเปกตรัมพลังงานเสร็จสิ้น และถ้าต้องการกลับไปยังโหมด Gamma spec ให้กดปุ่ม B



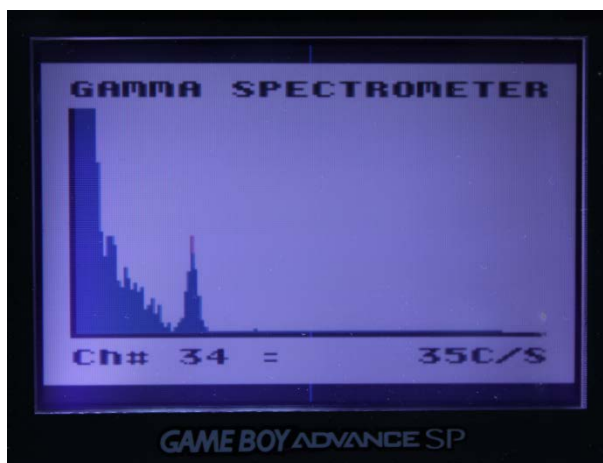
ภาพที่ 12

8. โหมด Sweep spectrometer เป็นโหมดสำหรับการสวீปสเปกตรัมพลังงานรังสี ผู้ใช้สามารถกำหนดค่าจำนวนรอบในการสวี่พ ดังภาพที่ 13 โดยการกดปุ่ม UP และปุ่ม DOWN ในการเพิ่มและลดค่า และกดปุ่ม Select เพื่อส่งค่าที่กำหนด



ภาพที่ 13

9. หลังจากกำหนดค่าจำนวนรอบในการสวீป และกดปุ่ม Select จะปรากฏหน้าจอตั้งภาพที่ 14 แสดงสเปกตรัมพลังงานรังสีที่ทำการวัดและสามารถเลื่อนเคอร์เซอร์โดยการกดปุ่ม LEFT หรือปุ่ม RIGHT เพื่ออ่านค่าจำนวนนับต่อเวลาที่ได้ในแต่ละช่องพลังงาน (Ch#) และสามารถขยายและลดหน้าจอในแนวสูง โดยการกดปุ่ม UP หรือปุ่ม DOWN เมื่อวิเคราะห์สเปกตรัมพลังงานเสร็จสิ้น และถ้าต้องการกลับไปยังโหมด Gamma spec ให้กดปุ่ม B



ภาพที่ 14

ภาคผนวก ข

โปรแกรมควบคุมและแสดงผลของเครื่องแกมมาสเปกโตรมิเตอร์

โปรแกรมควบคุมและแสดงผลสำหรับไมโครคอนโทรลเลอร์ PIC16F877

```

Device 16F877
Xtal = 20
Remarks On
Reminders OFF
Reminders On
On_Interrupt Horloge
Disable Interrupt
All_Digital On
**** Declare Symbol for Interrupt ***
Symbol GIE = INTCON.7
Symbol T0IF = INTCON.2
Symbol T0IE = INTCON.5
Symbol PS0 = OPTION_REG.0
Symbol PS1 = OPTION_REG.1
Symbol PS2 = OPTION_REG.2
Symbol PSA = OPTION_REG.3
Symbol T0SE= OPTION_REG.4
Symbol T0CS= OPTION_REG.5
Symbol TMR1_ON T1CON.0
Symbol INT_RTIF PIR1.0
Symbol INT_RTIE PIE1.0
**** Declare control pins of Buttom Switch and Others ***
Symbol bit1 PORTE.0
Symbol bit2 PORTE.1
Symbol bit3 PORTE.2
**** Declare control pins for sca ***
Symbol _cs PORTC.7
Symbol _wr PORTC.6
Symbol a0 PORTC.5
Symbol setmode PORTC.4
TRISB = %00000000
TRISC.4 = 0
TRISC.5 = 0
TRISC.6 = 0
TRISC.7 = 0
TRISD = %00000000
PORTD = %00000000
**** Declare variables for Counter *****

Dim No_Count As Word
Dim i As Word
Dim VAL_HORLOGE As Word
Dim Limite_Horloge As Word
Dim SECONDE As Byte
Dim chk As Bit
Dim FT As Bit
Dim TempE As Byte
Dim TempB As Byte
**** Declare variables for sca *****
Dim lld As Byte
Dim uld As Byte
Dim win As Byte
Dim time As Byte
Dim lmax As Byte
Dim lloop As Byte
Dim roun As Byte
Dim dweel_t As Byte
Dim coun_sweep[128] As Byte
Dim c As Byte
Dim ii As Byte
Dim r As Byte
Dim dataout As Word
****Start Program here ****
Initialisation:
T0SE =0
T0CS =0
PSA =0
PS0 =1
PS1 =1
PS2 =0
GIE=0
Limite_Horloge= 125
VAL_HORLOGE=0
i=0

Init_Program:
INTCON = %10000000

```

```

T1CON.1 = 1 : T1CON.2 = 1 : T1CON.3 = 0
T1CON.4 = 0 : T1CON.5 = 0
TMR1_ON = 0
TMR1H = 0
TMR1L = 0
INTCON.5 = 0
SelectMode:
  TRISE = %00000111
  TempE = PORTE
  If TempE = 1 Then
    chk = 1
    lld = 15
    time = 10
    GoTo SetRate
  Elseif TempE = 2 Then
    chk = 1
    GoTo Spectrum
  Elseif TempE = 3 Then
    chk = 1
    GoTo round
  EndIf
  GoTo SelectMode
RateMeter:
  TRISE = %00000111
  TempE = PORTE
  If TempE = 3 Then
    TRISB = %11111111
    lld = PORTB
    GoTo SetRate
  EndIf
  GoTo RateMeter
SetRate:
  uld = 255
  _cs = 1 : _wr = 1
  TRISD = %00000000
  PORTD = %00000000
  a0 = 0
  DelayUS 100
  _cs = 0 : _wr = 0
  PORTD = lld
  DelayUS 100
  _wr = 1
  DelayUS 100
  _cs = 1
  GoTo StartCount
*****For Spectrometer*****
Spectrum:
  TRISE = %00000111
  TempE = PORTE
  If TempE = 3 Then
    TRISB = %11111111
    lld = PORTB
    GoTo Resivewin
  EndIf
  GoTo Spectrum
*****For Spectrometer*****
Resivewin:
  TRISE = %00000111
  TempE = PORTE
  If TempE = 4 Then
    TRISB = %11111111
    win = PORTB
    GoTo Resivetime
  EndIf
  GoTo Resivewin
*****For Spectrometer*****
Resivetime:
  TRISE = %00000111
  TempE = PORTE
  If TempE = 5 Then
    TRISB = %11111111

```

```

time = PORTB
time = time * 10
lmax = ((255-lld)/win)
lloop = 0
GoTo Setsca
Endlf
GoTo Resivetime
*****For Spectrometer*****
Setsca:
_cs = 1 : _wr = 1
uld = lld + win
Inc lloop
TRISD = %00000000
PORTD = %00000000
a0 = 0
DelayUS 100
_cs = 0 : _wr = 0
PORTD = lld
DelayUS 100
_wr = 1
DelayUS 100
_cs = 1
DelayUS 100
a0 = 1
DelayUS 100
_cs = 0 : _wr = 0
PORTD = uld
DelayUS 100
_wr = 1
DelayUS 100
_cs = 1
GoTo StartCountsca
Enable Interrupt
*****For Ratemeter*****
StartCount:
FT = 0
If chk = 1 Then
No_Count = 0
TMR1H = 0
TMR1L = 0
SECONDE = time
INTCON.5 = 1
T0IF=1
GoTo Count_Loop
Endlf
GoTo StartCount
*****For Spectrometer*****
StartCountsca:
FT = 0
If chk = 1 Then
No_Count = 0
TMR1H = 0
TMR1L = 0
SECONDE = time
INTCON.5 = 1
T0IF=1
GoTo Count_Loopsca
Endlf
GoTo StartCountsca
*****For Ratemeter*****
Count_Loop:
TMR1_ON = 1
If FT = 1 Then
No_Count.LowByte = TMR1L
No_Count.HighByte = TMR1H
TRISE = %00000000
Low bit1 : Low bit2 : Low bit3
TRISB = %00000000
PORTB = No_Count.LowByte
i=0
GoTo sent_low
Endlf
GoTo Count_Loop
*****For Spectrometer*****
Count_Loopsca:
TMR1_ON = 1
If FT = 1 Then
No_Count.LowByte = TMR1L
No_Count.HighByte = TMR1H
TRISE = %00000000

```

```

        Low bit1 : Low bit2 : Low bit3
        TRISB = %00000000
        PORTB = No_Count.LowByte
        i=0
        GoTo sent_lowscas
    EndIf
    GoTo Count_Loopsca
*****For Ratemeter*****
sent_low:
    TRISE = %00000000
    If i = 100 Then
        High bit1 : Low bit2 : Low bit3
        Inc i
        GoTo sent_low
    ElseIf i = 160 Then
        i = 0
        TRISB = %00000000
        PORTB = No_Count.HighByte
        High bit1 :High bit2 : Low bit3
        GoTo sent_high
    Else
        Inc i
        GoTo sent_low
    EndIf
*****For Spectormeter*****
sent_lowscas:
    TRISE = %00000000
    If i = 100 Then
        High bit1 : Low bit2 : Low bit3
        Inc i
        GoTo sent_lowscas
    ElseIf i = 160 Then
        i = 0
        TRISB = %00000000
        PORTB = No_Count.HighByte
        High bit1 :High bit2 : Low bit3
        GoTo sent_highscas
    Else
        Inc i
        GoTo sent_lowscas
    EndIf
*****For Ratemeter*****
sent_high:
    TRISE = %00000000
    If i = 100 Then
        Low bit1 : High bit2 : Low bit3
        Inc i
        GoTo sent_high
    ElseIf i = 160 Then
        TRISE = %00000000
        Low bit1 : Low bit2 : Low bit3
        TRISE = %00000111
        TempE = PORTE
        If TempE = 6 Then
            GoTo SelectMode
        Else
            GoTo StartCount
        EndIf
    Else
        Inc i
        GoTo sent_high
    EndIf
*****For Spectormeter*****
sent_highscas:
    TRISE = %00000000
    If i = 100 Then
        Low bit1 : High bit2 : Low bit3
        Inc i
        GoTo sent_highscas
    ElseIf i = 200 Then
        TRISE = %00000000
        Low bit1 : Low bit2 : Low bit3
        lld = uld
        If lloop = lmax Then
            GoTo SelectMode
        Else
            GoTo Setsca
        EndIf
    Else
        Inc i

```

```

        GoTo sent_highsca
    EndIf
*****For Sweep Mode*****
round:
    TRISE = %00000111
    TempE = PORTE
    If TempE = 2 Then
        TRISB = %11111111
        roun = PORTB
        lld = 0 : win = 2 : lmax = 127 : r = 1 : lloop=0
        DelayMS 900
        GoTo Set_sweep
    EndIf
    GoTo round
*****set new to sweep*****
chk_round:
    If r = roun Then
        DelayMS 1400
        TRISE = %00000111
        TempE = PORTE
        GoTo SelectMode
    Else
        lloop = 0 : lld = 0 : uld = 0
        Inc r
        DelayMS 1600
        GoTo Set_sweep
    End If
Set_sweep:
    _cs = 1 : _wr = 1
    uld = lld + win
    TRISD = %00000000
    PORTD = %00000000
    a0 = 0
    DelayUS 100
    _cs = 0 : _wr = 0
    PORTD = lld
    DelayUS 100
    _wr = 1
    DelayUS 100
    _cs = 1
    DelayUS 100
    a0 = 1
    DelayUS 100
    _cs = 0 : _wr = 0
    PORTD = uld
    DelayUS 100
    _wr = 1
    DelayUS 100
    _cs = 1
    GoTo StartswEEP
Enable Interrupt
StartswEEP:
    FT = 0
    If chk = 1 Then
        No_Count = 0
        TMR1H = 0
        TMR1L = 0
        SECONDE = 1
        INTCON.5 = 1
        T0IF=1
        GoTo Count_sweep
    EndIf
    GoTo StartswEEP
Count_sweep:
    TMR1_ON = 1
    If FT = 1 Then
        No_Count.LowByte = TMR1L
        No_Count.HighByte = TMR1H
        TRISE = %00000000
        Low bit1 : Low bit2 : Low bit3
        TRISB = %00000000
        PORTB = No_Count.LowByte
        High bit1 : Low bit2 : Low bit3
        i=0
        GoTo sent_lowswEEP
    EndIf
    GoTo Count_sweep
sent_lowswEEP:
    TRISE = %00000000
    If i >= 1 And i < 29 Then

```

```

High bit1 : Low bit2 : Low bit3
Inc i
GoTo sent_lowswEEP
ElseIf i = 29 Then
  i = 0
  TRISB = %00000000
  Low bit1 : Low bit2 : Low bit3
  lld = uld
  If lloop = lmax Then
    GoTo chk_round
  Else
    Inc lloop
    GoTo Set_swEEP
  End If
Else
  Inc i
  GoTo sent_lowswEEP
EndIf

```

```

'-----
' INTERRUPT ROUTINE
'-----
Horloge:
  Inc VAL_HORLOGE
  If VAL_HORLOGE > Limite_Horloge Then
    VAL_HORLOGE=0
    SECONDE=SECONDE-1
    If SECONDE = 0 Then
      TMR1_ON = 0 : FT = 1
      INTCON.5 = 0 : 'INT_RTIE = 0
    EndIf
  EndIf
  TMR0 = 6
  T0IF=0
  Context Restore
End

```

โปรแกรมฟังก์ชันการทำงานสำหรับไอซีพีเอสดี 813F2

```

#ifndef PSD813F2_H
#define PSD813F2_H
#include <gb\gb.h>
#define CSIOP 0xB000
/*****
PSD813F2 registers
*/
#define CONTROL_A (*(__REG)(CSIOP+0x02))
#define DIRECTION_A (*(__REG)(CSIOP+0x06))
#define DRIVE_A (*(__REG)(CSIOP+0x08))
// MCU I/O mode is selected in PSDsoft Express
// so Output Enable bit is ignored
#define OUTENABLE_A (*(__REG)(CSIOP+0x0C))
#define DATAIN_A (*(__REG)(CSIOP+0x00))
#define DATAOUT_A (*(__REG)(CSIOP+0x04))
#define CONTROL_B (*(__REG)(CSIOP+0x03))
#define DIRECTION_B (*(__REG)(CSIOP+0x07))
#define DRIVE_B (*(__REG)(CSIOP+0x09))
// MCU I/O mode is selected in PSDsoft Express
// so Output Enable bit is ignored
#define OUTENABLE_B (*(__REG)(CSIOP+0x0D))
#define DATAIN_B (*(__REG)(CSIOP+0x01))
#define DATAOUT_B (*(__REG)(CSIOP+0x05))
#define DIRECTION_C (*(__REG)(CSIOP+0x14))
#define DRIVE_C (*(__REG)(CSIOP+0x16))
// MCU I/O mode is selected in PSDsoft Express
// so Output Enable bit is ignored
#define OUTENABLE_C (*(__REG)(CSIOP+0x1A))
#define DATAIN_C (*(__REG)(CSIOP+0x10))
#define DATAOUT_C (*(__REG)(CSIOP+0x12))
#define DIRECTION_D (*(__REG)(CSIOP+0x15))
#define DRIVE_D (*(__REG)(CSIOP+0x17))
// MCU I/O mode is selected in PSDsoft Express
// so Output Enable bit is ignored
#define OUTENABLE_D (*(__REG)(CSIOP+0x1B))
#define DATAIN_D (*(__REG)(CSIOP+0x11))
#define DATAOUT_D (*(__REG)(CSIOP+0x13))
#define IMC_A (*(__REG)(CSIOP+0x0A))

```

```

#define IMC_B          (*(__REG)(CSIOP+0x0B))
#define IMC_C          (*(__REG)(CSIOP+0x18))
#define OMC_AB        (*(__REG)(CSIOP+0x20))
#define OMC_BC        (*(__REG)(CSIOP+0x21))
#define OMCMASK_AB    (*(__REG)(CSIOP+0x22))
#define OMCMASK_BC    (*(__REG)(CSIOP+0x23))
#define MAINPROTECT   (*(__REG)(CSIOP+0xC0))
#define ALTPROTECT    (*(__REG)(CSIOP+0xC2))
#define JTAG           (*(__REG)(CSIOP+0xC7))
#define PMMR0         (*(__REG)(CSIOP+0xB0))

#define PMMR2          (*(__REG)(CSIOP+0xB4))
#define PAGE          (*(__REG)(CSIOP+0xE0))
#define VM            (*(__REG)(CSIOP+0xE2))
#define APD_ENABLEor  0x02
#define PLD_LOW_POWERor 0x08
#define PLD_HI_POWERand 0xF7
#define PLD_ARRAY_CLKand 0x10
#define PLD_MCELL_CLKand 0x20
/*****
#endif // PSD813F2_H

```

โปรแกรมควบคุมและแสดงผลสำหรับเครื่องเล่นเกมบอย

```

#include <gb/gb.h>
#include <gb/cgb.h>
#include <stdio.h>
#include <gb/drawing.h>
#include <gb/console.h>
#include "psd813f2.h"
#include <cursor2.h>
#include <cursor2.c>
#define WHITE 0
#define LTGREY 1
#define DKGREY 2
#define BALAK 3
#define M_NOFILL 0
#define M_FILL 1
UBYTE c1,c2,c3,c4,c5,s1,s2,s3,s4,s5,chkselect;
UBYTE i,j,l,ll,loop,ld, roun, r;
UBYTE low_count,high_count,sca,lmax,
sweep_count[128];
UWORD sa,datacal;
UINT8 scay,time,win;
UWORD datasca[128], datasca1[128];
unsigned int curx;
void rate_meter();
void modeselect();
void receive_count();
void send_lld();

void mode_spec();
void clear()
{
    int x,y;
    for (y=0; y<18; y++)
        for (x=0; x<20; x++)
            {
                gotogxy(x,y);
                wrtchr(' ');
            }
    gotogxy(0,0);
}
UWORD obj_p[] =
{
    cursorCGBPAl0c0,cursorCGBPAl0c1,cursorC
    GBPAl0c2,cursorCGBPAl0c3,
    cursorCGBPAl1c0,cursorCGBPAl1c1,cursorC
    GBPAl1c2,cursorCGBPAl1c3
};
void init_palette()
{
    set_sprite_palette( 0, 1, &obj_p[6] );
    set_bkg_palette( 0, 1, &obj_p[10] );
}
void init()
{

```

```

clear();
gotogxy(7,2);color(3,0,SOLID);gprintf("GAMEB
OY");
gotogxy(1,4);gprintf("GAMMA
SPECTROMETER");
gotogxy(3,6);color(2,0,SOLID);gprintf("DEVEL
OPMENT BY");
gotogxy(0,8);gprintf("PIYATHEP
CHOBTHUMKIT");
gotogxy(2,10);color(3,0,SOLID);gprintf("DATE
29-01-2555");
gotogxy(4,12);color(2,0,SOLID);gprintf("TEST
Program");
gotogxy(7,14);color(2,0,SOLID);gprintf("SCA_1
4");
gotogxy(2,17);color(3,0,SOLID);gprintf("Please
any button");
waitpad(0xFF);
waitpadup();
}
//this is test data for each x1
unsigned int datax1[]={
    33,33,34,35,35,36,37,38,39,39,40,41,42,43,44,
    44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,
    59,60,61,62,63,64,65,66,67,68,69,69,70,71,73,
    74,75,76,77,78,79,80,81,82,83,84,85,87,88,89,
    91,92,93,94,95,96,97,98,99,100,101,102,103,1
    04,105,106,107,108,109,110,111,112,113,114,
    115,116,116,117,118,119,120,121,121,122,12
    3,124,125,125,126,127,127
};
//this is test data for each x2
unsigned int datax2[]={
    33,33,34,35,35,36,37,38,39,39,40,41,42,43,44,
    44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,
    59,60,61,62,63,64,65,66,67,68,69,70,71,72,74,
    75,76,77,78,79,80,81,82,83,84,85,86,88,89,90,
    91,92,93,94,95,96,97,98,99,100,101,102,103,1
    04,105,106,107,108,109,110,111,112,113,114,
    115,116,116,117,118,119,120,121,121,122,12
    3,124,125,125,126,127,127
};
//this is test data for each y1
unsigned int datay1[]={
    61,60,60,59,58,57,56,56,55,54,54,53,52,51,51,
    50,50,49,48,48,47,47,46,46,45,45,44,44,44,43,
    43,42,42,42,41,41,41,41,40,40,40,40,40,38,38,
    38,38,38,38,38,38,38,38,38,38,38,38,38,38,
    38,38,38,39,39,39,39,40,40,40,41,41,42,42,42,
    43,43,44,44,45,45,46,46,47,48,48,49,49,50,51,
    52,52,53,54,54,55,56,57,58,58,59
};
//this is test data for each y2
unsigned int datay2[]={
    60,59,59,58,57,56,55,55,54,53,53,52,51,50,50,
    49,49,48,47,47,46,46,45,45,44,44,43,43,43,42,
    42,41,41,41,40,40,40,40,39,39,39,39,39,38,38,
    38,38,38,38,38,38,38,38,38,38,38,38,39,39,
    39,39,39,40,40,40,40,41,41,41,42,42,43,43,43,
    44,44,45,45,46,46,47,47,48,49,49,50,50,51,52,
    53,53,54,55,55,56,57,58,59,59,60
};
//this is test data for each count*1
unsigned int datacx1[]={
    0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,
    19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,
    34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,
    49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,
    64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,
    79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,
    94,95,96,97,98,99,100
};
// this simulate function for get_countx1 from rate meter
unsigned int get_countx1(unsigned int chax)
{
    return(datax1[chax]);
}
// this simulate function for get_countx2 from rate meter
unsigned int get_countx2(unsigned int chax)

```



```

{
    return(datax2[chax]);
}
// this simulate function for get_county1 from rate meter
unsigned int get_county1(unsigned int chay)
{
    return(datay1[chay]);
}
// this simulate function for get_county2 from rate meter
unsigned int get_county2(unsigned int chay)
{
    return(datay2[chay]);
}
// this simulate function for get_countcx1 from rate meter
unsigned int get_countcx1(unsigned int chac)
{
    return(datacx1[chac]);
}
void pl_meter(void)
{
    gotogxy(4,1);color(3,0,SOLID);gprintf("SERVE
Y METER");
    gotogxy(14,12);gprintf("mR/Hr");
    color(BLACK,WHITE,SOLID);
    gotogxy(1,6);gprintf("0");
    gotogxy(4,4);gprintf("20");
    gotogxy(7,3);gprintf("40");
    gotogxy(11,3);gprintf("60");
    gotogxy(14,4);gprintf("80");
    gotogxy(16,6);gprintf("100");
    color(DKGREY,WHITE,SOLID);
    /* Number line 0*/
    line(21,57,30,65);line(21,58,30,66);
    /* Number line 1*/
    line(35,54,38,57);
    /* Number line 2*/
    line(40,42,47,51);line(40,43,47,52);
    /* Number line 3*/
    line(55,42,57,46);
    /* Number line 4*/
    line(66,33,68,43);line(66,34,68,44);
    /* Number line 5*/
    line(80,37,80,42);

    /* Number line 6*/
    line(94,33,92,43);line(94,34,92,44);
    /* Number line 7*/
    line(105,42,103,46);
    /* Number line 8*/
    line(120,42,113,51);line(120,43,113,52);
    /* Number line 9*/
    line(125,54,122,57);
    /* Number line 10*/
    line(139,57,130,65);line(139,58,130,66);
    //line up
    line(2,17,157,17);line(2,18,157,18);line(2,19,15
7,19);
    //line left
    line(2,17,2,107);line(3,17,3,107);line(4,17,4,10
7);
    //line right
    line(155,17,155,107);line(156,17,156,107);line(
157,17,157,107);
    //line down
    line(2,105,157,105);line(2,106,157,106);line(2,
107,157,107);
    line(2,108,157,108);line(2,109,157,109);
    //circle center
    color(LTGREY,DKGREY,SOLID);circle(80,107,
7,M_FILL);
}
void plot_meter() //โครงเพิ่ม
{
    color(BLACK,WHITE,SOLID);
    line(get_countx1(l),get_county1(l)+7,79,107);
    line(get_countx2(l),get_county2(l)+7,80,107);
    color(LTGREY,DKGREY,SOLID);
    circle(80,107,7,M_FILL);
    color(BLACK,WHITE,SOLID);
}

```

```

void disp()
{
    ll = l;
}

void calculus()
{
    UBYTE cou_poit;
    datasca[3]=datasca[7];
    datasca[7] = ((high_count*256)+low_count);
    datasca[1] = datasca[3]+datasca[7];
    datasca[1] = datasca[1]/2;
    if (datasca[1] == 0)
    {
        datasca[2] = 0;
    }else if ((datasca[1]>=1) &&
(datasca[1]<320))
    {
        datasca[2] = (datasca[1]*10)/65;
        s3=datasca[2]/100;
        s4=(datasca[2]%100)/10;
        s5=(datasca[2]%10)/1;
    }else if ((datasca[1]>=320) &&
(datasca[1]<830))
    {
        datasca[2] = datasca[1]/6;
        s3=datasca[2]/100;
        s4=(datasca[2]%100)/10;
        s5=(datasca[2]%10)/1;
    }else if ((datasca[1]>=830) &&
(datasca[1]<1100))
    {
        datasca[2] = datasca[1]/5;
        s3=datasca[2]/100;
        s4=(datasca[2]%100)/10;
        s5=(datasca[2]%10)/1;
    }else if ((datasca[1]>=1100) &&
(datasca[1]<2200))
    {
        datasca[2] = (datasca[1]*10)/48;
        s3=datasca[2]/100;

```

```

        s4=(datasca[2]%100)/10;
        s5=(datasca[2]%10)/1;
    }else if ((datasca[1]>=2200) &&
(datasca[1]<2900))
    {
        datasca[2] = datasca[1]/5;
        s3=datasca[2]/100;
        s4=(datasca[2]%100)/10;
        s5=(datasca[2]%10)/1;
    }else if ((datasca[1]>=2900) &&
(datasca[1]<3650))
    {
        datasca[2] = (datasca[1]*10)/41;
        s3=datasca[2]/100;
        s4=(datasca[2]%100)/10;
        s5=(datasca[2]%10)/1;
    }else if ((datasca[1]>=3650) &&
(datasca[1]<5080))
    {
        datasca[2] = (datasca[1]*10)/36;
        s2=datasca[2]/1000;
        s3=(datasca[2]%1000)/100;
        s4=(datasca[2]%100)/10;
        s5=(datasca[2]%10)/1;
    }else if ((datasca[1]>=5080) &&
(datasca[1]<6000))
    {
        datasca[2] = (datasca[1]*10)/32;
        s2=datasca[2]/1000;
        s3=(datasca[2]%1000)/100;
        s4=(datasca[2]%100)/10;
        s5=(datasca[2]%10)/1;
    }else if ((datasca[1]>=6000) &&
(datasca[1]<9500))
    {
        datasca[2] = (datasca[1]/31)*10;
        s2=datasca[2]/1000;
        s3=(datasca[2]%1000)/100;
        s4=(datasca[2]%100)/10;
        s5=(datasca[2]%10)/1;
    }

```

```

}else if ((datasca[1]>=9500) &&
(datasca[1]<14000))
{
    datasca[2] = (datasca[1]/28)*10;
    s2=datasca[2]/1000;
    s3=(datasca[2]%1000)/100;
    s4=(datasca[2]%100)/10;
    s5=(datasca[2]%10)/1;
}else if ((datasca[1]>=14000) &&
(datasca[1]<22000))
{
    datasca[2] = (datasca[1]/22)*10;
    s1=datasca[2]/10000;
    s2=(datasca[2]%10000)/1000;
    s3=(datasca[2]%1000)/100;
    s4=(datasca[2]%100)/10;
    s5=(datasca[2]%10)/1;
}else if (datasca[1]>=22000)
{
    datasca[2] = datasca[1]/2;
    s1=datasca[2]/10000;
    s2=(datasca[2]%10000)/1000;
    s3=(datasca[2]%1000)/100;
    s4=(datasca[2]%100)/10;
    s5=(datasca[2]%10)/1;
}
c1=datasca[1]/10000;
c2=(datasca[1]%10000)/1000;
c3=(datasca[1]%1000)/100;
c4=(datasca[1]%100)/10;
c5=(datasca[1]%10)/1;
color(WHITE,WHITE,SOLID);
line(get_countx1(ll),get_county1(ll)+7,79,107);
line(get_countx2(ll),get_county2(ll)+7,80,107);
color(LTGREY,DKGREY,SOLID);
circle(80,107,7,M_FILL);
if ((datasca[2] >= 0) && (datasca[2] <= 99))
{
    cou_poit = ((s4*10)+(s5));
    l = cou_poit;
    color(BLACK,WHITE,SOLID);
    plot_meter();
    color(3,0,SOLID);
    if (chkselect == 1)
    {
        color(3,0,SOLID);
        gotogxy(1,16);gprintf("
0.%u%u mR/Hr ",s4,s5);
    }else
    {
        color(3,0,SOLID);
        if (c3>=1)
        {
            gotogxy(1,16);gprintf("
%u%u%u CPS ",c3,c4,c5);
        }else if (c4>=1)
        {
            gotogxy(1,16);gprintf("
%u%u CPS ",c4,c5);
        }else if (c5>=0 && c4<1)
        {
            gotogxy(1,16);gprintf("    %u
CPS ",c5);
        }
    }
    gotogxy(1,12);
    gprintf("x0.01");
}else if ((datasca[2] >= 100) && (datasca[2]
<= 999))
{
    cou_poit = ((s3*10)+(s4));
    l = cou_poit;
    plot_meter();
    color(3,0,SOLID);
    if (chkselect == 1)
    {
        color(3,0,SOLID);
        gotogxy(1,16);gprintf("
%u.%u%u mR/Hr ",s3,s4,s5);
    }else

```

```

{
    color(3,0,SOLID);
    if (c2>=1)
    {
        gotogxy(1,16);
        gprintf(" %u,%u%u%u CPS
",c2,c3,c4,c5);
    }else if (c3>=1)
    {
        gotogxy(1,16);gprintf("
%u%u%u CPS ",c3,c4,c5);
    }else if (c4>=1)
    {
        gotogxy(1,16);gprintf("
%u%u CPS ",c4,c5);
    }
}
gotogxy(1,12);
gprintf("x0.1 ");
}else if ((datasca[2] >= 1000) && (datasca[2]
<= 9999))
{
    cou_poit = ((s2*10)+(s3));
    l = cou_poit;
    plot_meter();
    color(3,0,SOLID);
    if (chkselect == 1)
    {
        color(3,0,SOLID);
        gotogxy(1,16);
        gprintf(" %u%u.%u%u mR/Hr
",s2,s3,s4,s5);
    }else
    {
        color(3,0,SOLID);
        if (c1>=1)
        {
            gotogxy(1,16);
            gprintf(" %u%u,%u%u%u CPS
",c1,c2,c3,c4,c5);
        }else if (c2>=1)
        {
            gotogxy(1,16);
            gprintf(" %u,%u%u%u CPS
",c2,c3,c4,c5);
        }else if (c3>=1)
        {
            gotogxy(1,16);gprintf("
%u%u%u CPS ",c3,c4,c5);
        }
    }
    gotogxy(1,12);
    gprintf("x1 ");
}else
{
    cou_poit = ((s1*10)+(s2));
    l = cou_poit;
    plot_meter();
    color(3,0,SOLID);
    if (chkselect == 1)
    {
        color(3,0,SOLID);
        gotogxy(1,16);
        gprintf(" %u%u%u.%u%u mR/Hr
",s1,s2,s3,s4,s5);
    }else
    {
        color(3,0,SOLID);
        if (c1>=1)
        {
            gotogxy(1,16);
            gprintf(" %u%u,%u%u%u CPS
",c1,c2,c3,c4,c5);
        }else if (c2>=1)
        {
            gotogxy(1,16);
            gprintf(" %u,%u%u%u CPS
",c2,c3,c4,c5);
        }
    }
}

```

```

        gotogxy(1,12);
        gprintf("x10 ");
    }
}
void receive_count()
{
    DIRECTION_D = 0x00U;
    DATAIN_D = (DATAIN_D & 0xF700);
    DIRECTION_B = 0x00U;
    DATAIN_B = DATAIN_B & 0xBF00;
    i = 0;
    j = 0;
    while (i == 0)
    {
        j = DATAIN_D;
        switch (j)
        {
            case 1:
                break;
            case 2:
                high_count = DATAIN_B;
                i = 1;
                break;
            case 3:
                high_count = DATAIN_B;
                break;
            case 17:
                break;
            case 18:
                high_count = DATAIN_B;
                i = 1;
                break;
            case 19:
                high_count = DATAIN_B;
                break;
            default:
                low_count = DATAIN_B;
                break;
        }
    }
}
}
//***** Mode Rate meter *****
void rate_meter()
{
    UINT8 key;
    clear();
    color(3,0,SOLID);
    gotogxy(4,1);gprintf("SURVEY METER");
    clear();
    pl_meter();
    plot_meter();
    do
    {
        receive_count();
        calculus();
        disp();
    }while (!(joypad()));
wait:
    key = waitpad(J_START | J_B | J_UP
|J_DOWN);
    waitpadup();
    if (key&J_START)
    {
        DIRECTION_D = 0x07U;
        DATAOUT_D = 0x06U;
        DIRECTION_D = 0x07U;
        DATAOUT_D = 0x01U;
        rate_meter();
    }else if (key&J_B)
    {
        DIRECTION_D = 0x07U;
        DATAOUT_D = 0x06U;
        modeselect();
    }else if (key&J_UP)
    {
        chkselect = 1;
        rate_meter();
    }else if (key&J_DOWN)
    {
        chkselect = 2;

```

```

        rate_meter();
    }else
    {
        goto wait;
    }
}
//*****plot line x,y FOR SPECTRUM*****
void plot_spec()
{
    clear();
    color(3,0,SOLID);
    gotogxy(1,1);gprintf("GAMMA
SPECTROMETER");
    color(3,0,SOLID);
    //*****y
    line(8,20,8,122);line(9,20,9,122);
    //*****x
    line(10,121,150,121);line(10,122,150,122);
}
//*****send parameter for spectrum*****
void send_llid()
{
    UINT8 llidset;UBYTE llid0;UBYTE llid1;UBYTE
llid2;UWORD llidcal;
    llid = 0;llid0 = 0;llid1 = 0;llid2 = 0;
    gotogxy(3,6);color(3,0,SOLID);
    gprintf("LLD = %u.%u%u V ",llid2,llid1,llid0);
    gotogxy(0,16);color(2,0,SOLID);
    gprintf("**Up Or Down");
    gotogxy(2,17);color(2,0,SOLID);
    gprintf("And Then SELECT");
wait:
    llidset = waitpad(J_UP | J_DOWN | J_SELECT);
    waitpadup();
    if(llidset&J_UP)
    {
        gotogxy(3,6);
        color(3,0,SOLID);
        llid=llid+1;
        llidcal=llid;
        llidcal=llidcal*4;
        llid0=llidcal%10;
        llid1=(llidcal%100)/10;
        llid2=llidcal/100;
        gprintf("LLD = %u.%u%u V
",llid2,llid1,llid0);
        goto wait;
    }else if(llidset&J_DOWN)
    {
        gotogxy(3,6);
        color(3,0,SOLID);
        llid=llid-1;
        llidcal=llid;
        llidcal=llidcal*4;
        llid0=llidcal%10;
        llid1=(llidcal%100)/10;
        llid2=llidcal/100;
        gprintf("LLD = %u.%u%u V
",llid2,llid1,llid0);
        goto wait;
    }else if(llidset&J_SELECT)
    {
        DIRECTION_B = 0xFFU;
        DATAOUT_B = llid;
        gotogxy(3,8);
        gprintf("dataout_b = %u mV
",DATAOUT_B);
        DIRECTION_D = 0x07U;
        DATAOUT_D = 0x03U;
        gotogxy(3,7);
        gprintf("dataout_d = %u
",DATAOUT_D);
    }else
    {
        goto wait;
    }
}
void send_window()
{

```

```

UINT8 winset;UBYTE win0;UBYTE
win1;UBYTE win2;UWORD wincal;
win = 0;win0 = 0;win1 = 0;win2 = 0;
clear();
color(3,0,SOLID);
gotogxy(1,1);gprintf("GAMMA
SPECTROMETER");
gotogxy(3,6);
gprintf("WIN = %u.%u%u V ",win2,win1,win0);
gotogxy(0,16);color(2,0,SOLID);
gprintf("**Up Or Down");
gotogxy(2,17);
gprintf("And Then SELECT");

wait:
winset = waitpad(J_UP | J_DOWN |
J_SELECT);
waitpadup();
if(winset&J_UP)
{
    gotogxy(3,6);color(3,0,SOLID);
    win=win+1;
    wincal=win;
    wincal=wincal*4;
    win0=wincal%10;
    win1=(wincal%100)/10;
    win2=wincal/100;
    gprintf("WIN = %u.%u%u V
    ",win2,win1,win0);
    goto wait;
}else if(winset&J_DOWN)
{
    gotogxy(3,6);color(3,0,SOLID);
    win=win-1;
    wincal=win;
    wincal=wincal*4;
    win0=wincal%10;
    win1=(wincal%100)/10;
    win2=wincal/100;
    gprintf("WIN = %u.%u%u V
    ",win2,win1,win0);
    goto wait;
}else if(winset&J_SELECT)
{
    DIRECTION_B = 0xFFU;
    DATAOUT_B = win;
    gotogxy(3,8);
    gprintf("dataout_b = %u mV
    ",DATAOUT_B);
    DIRECTION_D = 0x07U;
    DATAOUT_D = 0x04U;
    gotogxy(3,7);
    gprintf("dataout_d = %u
    ",DATAOUT_D);
}else
{
    goto wait;
}
}

void send_time()
{
    UINT8 timeset;time = 0;
    clear();
    color(3,0,SOLID);
    gotogxy(1,1);gprintf("GAMMA
    SPECTROMETER");
    gotogxy(3,6);
    gprintf("Time = %u sec ",time);
    gotogxy(0,16);color(2,0,SOLID);
    gprintf("**Up Or Down");
    gotogxy(2,17);
    gprintf("And Then SELECT");

    wait:
    timeset = waitpad(J_UP | J_DOWN |
    J_SELECT);
    waitpadup();
    if(timeset&J_UP)
    {
        gotogxy(3,6);color(3,0,SOLID);
        time++;
        gprintf("Time = %u sec ",time);
    }
}

```

```

        goto wait;

    }else if(timeset&J_DOWN)
    {
        gotogxy(3,6);color(3,0,SOLID);
        time--;
        gprintf("Time = %u sec ",time);
        goto wait;
    }else if(timeset&J_SELECT)
    {
        DIRECTION_B = 0xFFU;
        DATAOUT_B = time;
        gotogxy(3,8);
        gprintf("dataout_b = %u sec
        ",DATAOUT_B);
        DIRECTION_D = 0x07U;
        DATAOUT_D = 0x05U;
        gotogxy(3,7);
        gprintf("dataout_d = %u
        ",DATAOUT_D);
    }else
    {
        goto wait;
    }
}
void calasca()
{
    UINT8 scai;
    datasca[scai]=(high_count*256)+low_count;
    if (datacal<datasca[scai])
    {
        datacal = datasca[scai];
    }else
        datacal = datacal;
    if (datacal>=51201)
    {
        sa = 1024;
    }else if (datacal>=25601 && datacal<51200)
    {
        sa = 512;
    }else if (datacal>=12801 && datacal<25600)
    {
        sa = 256;
    }else if (datacal>=6401 && datacal<12800)
    {
        sa = 128;
    }else if (datacal>=3201 && datacal<6400)
    {
        sa = 64;
    }else if (datacal>=1601 && datacal<3200)
    {
        sa = 32;
    }else if (datacal>=801 && datacal<1600)
    {
        sa = 16;
    }else if (datacal>=401 && datacal<800)
    {
        sa = 8;
    }else if (datacal>=201 && datacal<400)
    {
        sa = 4;
    }else if (datacal>=101 && datacal<200)
    {
        sa = 2;
    }else
        sa = 1;
    c1=datasca[scai]/10000;
    c2=(datasca[scai]%10000)/1000;
    c3=(datasca[scai]%1000)/100;
    c4=(datasca[scai]%100)/10;
    c5=(datasca[scai]%10);
    gotogxy(1,16);color(3,0,SOLID);
    if (lloop>=0 && lloop<10)
    {
        gprintf("Ch# %u ",lloop);
    }else if (lloop>=10 && lloop<100)
    {
        gprintf("Ch# %u ",lloop);
    }else
        gprintf("Ch#%u ",lloop);
}

```



```

if (datasca[sca]>=10000)
{
    gprintf("= %u%u,%u%u%uC/S
    ",c1,c2,c3,c4,c5);
}else if (datasca[sca]>999 &&
datasca[sca]<=9999)
{
    gprintf("= %u,%u%u%uC/S
    ",c2,c3,c4,c5);
}else if (datasca[sca]>99 &&
datasca[sca]<=999)
{
    gprintf("= %u%u%uC/S
    ",c3,c4,c5);
}else if (datasca[sca]>9 &&
datasca[sca]<=99)
{
    gprintf("= %u%uC/S ",c4,c5);
}else if (datasca[sca]>=0 &&
datasca[sca]<=9)
{
    gprintf("= %uC/S ",c5);
}
scai = sca;
for (sca=0;sca<127;sca++)
{
    color(2,0,SOLID);
    scay =120-(datasca[sca]/sa);
    line(sca+10,scay,sca+10,120);
}
sca = scai;
sca++;
}
void show_data(unsigned int x)
{
    c1=datasca[curx]/10000;
    c2=(datasca[curx]%10000)/1000;
    c3=(datasca[curx]%1000)/100;

```

```

c4=(datasca[curx]%100)/10;
c5=(datasca[curx]%10);
gotogxy(1,16);color(3,0,SOLID);
if (x>=0 && x<10)
{
    gprintf("Ch# %u ",x);
}else if (x>=10 && x<100)
{
    gprintf("Ch# %u ",x);
}else
    gprintf("Ch#%u ",x);
if (datasca[curx]>=10000)
{
    gprintf("= %u%u,%u%u%uC/S
    ",c1,c2,c3,c4,c5);
}else if (datasca[curx]>999 &&
datasca[curx]<=9999)
{
    gprintf("= %u,%u%u%uC/S
    ",c2,c3,c4,c5);
}else if (datasca[curx]>99 &&
datasca[curx]<=999)
{
    gprintf("= %u%u%uC/S
    ",c3,c4,c5);
}else if (datasca[curx]>9 &&
datasca[curx]<=99)
{
    gprintf("= %u%uC/S ",c4,c5);
}else if (datasca[curx]>=0 &&
datasca[curx]<=9)
{
    gprintf("= %uC/S ",c5);
}
}
//***** Mode Spectrum Move cursor *****
void move_cursor()
{
    curx=1;

```

```

        move_sprite(0,curx+11,128-
(datasca[curx]/sa));
        SHOW_SPRITES;
//start of infinite while loop
while(!(joypad()&J_B))
{
    if(joypad()&J_LEFT)
    {
        curx--; //update variable
        if(curx<=255)
        {
            curx=127;
        }
        if ((datasca[curx]/sa)>99)
        {
            scay = 100;
        }else
            scay =
(datasca[curx]/sa);
        move_sprite(0,curx+11,128-scay);
        delay(100);
    }
    if(joypad()&J_RIGHT)
    {
        curx++; //update variable
        if(curx>127)
        {
            curx=0;
        }
        if ((datasca[curx]/sa)>99)
        {
            scay = 100;
        }else
            scay =
(datasca[curx]/sa);
        move_sprite(0,curx+11,128-scay);
        delay(100); //delay
    }
    if(joypad()&J_UP)
    {

```

```

        sa = sa*2;
        if (sa>=2048)
        {
            sa=1024;
        }
        clear();
        plot_spec();
        for (sca=0;sca<128;sca++ )
            {
                color(2,0,SOLID);
                if ((datasca[sca]/sa)>99)
                {
                    scay = 20;
                }
                line(sca+10,scay,sca+10,120);
            }else
                scay =120-
                (datasca[sca]/sa);
            line(sca+10,scay,sca+10,120);
        }
        move_sprite(0,curx+11,128-(datasca[curx]/sa));
        delay(100); //delay
    }
    if(joypad()&J_DOWN)
    {
        sa = sa/2;
        if (sa==0)
        {
            sa=1;
        }
        clear();
        plot_spec();
        for (sca=0;sca<128;sca++ )
            {
                color(2,0,SOLID);
                if ((datasca[sca]/sa)>99)
                {
                    scay = 20;
                }
                line(sca+10,scay,sca+10,120);
            }
    }

```

```

        }else
        scay =120-
        (datasca[sca]/sa);
        line(sca+10,scay,sca+10,120);
        }
    move_sprite(0,curx+11,128-(datasca[curx]/sa));
    delay(100);
}
    show_data(curx);
} //end of infinite while loop
}
//***** Mode Spectrum SCAN *****/
void spectrum()
{
    UINT8 key;sca = 0;
    clear();
    datacal = 0;
    for (sca=0;sca<129;sca++)
    //clear sca array
    {
        datasca[sca] = 0;
    }
    clear();
    color(3,0,SOLID);
    gotogxy(1,1);gprintf("GAMMA
    SPECTROMETER");
    send_llid();
    send_window();
    send_time();
    plot_spec();
    lmax = ((255-llid)/win);
    do
    {
        lloop = lloop + 1;
        receive_count();
        calasca();
    }while(!(lmax == lloop));

    DIRECTION_D = 0x07U;
    DATAOUT_D = 0x06U;

```

```

    move_cursor();
wait:
    key = waitpad(J_START | J_B);
    waitpadup();
    if (key&J_START)
    {
        DIRECTION_D = 0x07U;
        DATAOUT_D = 0x02U;
        spectrum();
    }else if (key&J_B)
    {
        HIDE_SPRITES;
        modeselect();
    }else
    {
        goto wait;
    }
}
//*****set for sweep mode *****/
void scasweeptotal()
{
    clear();
    plot_spec();
    for (sca=0;sca<128;sca++)
    {
        color(2,0,SOLID);
        if ((datasca[sca]/sa)>99)
        {
            scay = 20;
            line(sca+10,scay,sca+10,120);
        }else
            scay =120-
            (datasca[sca]/sa);
            line(sca+10,scay,sca+10,120);
        }
    }
void calscasweep()
{
    datasca1[sca]=low_count;
    datasca[sca]=datasca1[sca]+datasca[sca];

```

```

if (datacal<datasca[sca])
{
    datacal = datasca[sca];
}else
    datacal = datacal;
sa = 8;
gotogxy(14,17);color(2,0,SOLID);
gprintf("%u  ",sca);
sca++;
}
void receive_sweep()
{
    DIRECTION_D = 0x00U;
    DATAIN_D = (DATAIN_D & 0xF700);
    DIRECTION_B = 0x00U;
    DATAIN_B = DATAIN_B & 0xBF00;
    i = 0;
    j = 0;
    while (i == 0)
    {
        j = DATAIN_D;
        switch (j)
        {
            case 1:
                low_count = DATAIN_B;
                i = 1;
                break;
            case 17:
                low_count = DATAIN_B;
                i = 1;
                break;
            default:
                low_count = DATAIN_B;
                break;
        }
    }
}
void send_round()
{
    UINT8 roundset;
    roun = 0;
    gotogxy(3,6);color(3,0,SOLID);
    gprintf("ROUND = %u  ",roun);
    gotogxy(0,16);color(2,0,SOLID);
    gprintf("**Up Or Down");
    gotogxy(2,17);color(2,0,SOLID);
    gprintf("And Then SELECT");
wait:
    roundset = waitpad(J_UP | J_DOWN |
    J_SELECT | J_B);
    waitpadup();
    if(roundset&J_UP)
    {
        gotogxy(3,6);
        color(3,0,SOLID);
        roun=roun+1;
        gprintf("ROUND = %u  ",roun);
        goto wait;
    }else if(roundset&J_DOWN)
    {
        gotogxy(3,6);
        color(3,0,SOLID);
        roun=roun-1;
        gprintf("ROUND = %u  ",roun);
        goto wait;
    }else if (roundset&J_B)
    {
        mode_spec();
    }else if(roundset&J_SELECT)
    {
        DIRECTION_B = 0xFFU;
        DATAOUT_B = roun;
        gotogxy(3,8);
        gprintf("dataout_b = %u
        ",DATAOUT_B);
        DIRECTION_D = 0x07U;
        DATAOUT_D = 0x02U;
        gotogxy(3,7);

```

```

        gprintf("dataout_d = %u
        ",DATAOUT_D);
    }else
    {
        goto wait;
    }
}
void sweep_mode()
{
    UINT8 key;sca = 0;
    clear();
    datacal = 0;
    for (sca=0;sca<129;sca++)
    {
        datasca[sca] = 0;
        datasca1[sca] = 0;
        lloop = 0;
    }
    clear();
    color(3,0,SOLID);
    gotogxy(1,1);gprintf("GAMMA
    SPECTROMETER");
    send_round();
    plot_spec();
    lmax = 128;
    for (r = 0; r<roun;r++ )
    {
        lloop = 0;
        sca = 0;
        gotogxy(2,16);color(2,0,SOLID);
        gprintf("waiting for sweep");
        gotogxy(5,17);color(2,0,SOLID);
        gprintf("round ");
        gotogxy(11,17);color(2,0,SOLID);
        gprintf("%u",r+1);
        do
        {
            lloop = lloop + 1;
            receive_sweep();
            calscasweep();
            }while(!(lmax == lloop));
            scasweeptotal();
        }
        DIRECTION_D = 0x07U;
        DATAOUT_D = 0x06U;
        move_cursor();
wait:
        key = waitpad(J_START | J_B);
        waitpadup();
        if (key&J_START)
        {
            DIRECTION_D = 0x07U;
            DATAOUT_D = 0x03U;
            sweep_mode();
        }else if (key&J_B)
        {
            HIDE_SPRITES;
            mode_spec();
        }else
        {
            goto wait;
        }
    }
}
void mode_spec()
{
    UINT8 choice,mode;
    clear();
    gotogxy(1,1);color(3,0,SOLID);
    gprintf("GAMMA SPECTROMETER");
    gotogxy(1,4);
    gprintf("MODE SELECT");
    gotogxy(2,6);color(2,0,SOLID);
    gprintf("SCAN MODE");
    color(3,0,1);
    box(12,44,90,58,0);
    gotogxy(2,8);color(2,0,SOLID);
    gprintf("SWEEP MODE");
    gotogxy(0,16);
    gprintf("**Up Or Down For MODE");
    gotogxy(2,17);

```

```

gprintf("And Then SELECT");
mode = 1;
wait:
choice = waitpad(J_UP | J_DOWN | J_SELECT
| J_B);
waitpadup();
if(choice&J_UP)
{
    color(3,0,1);
    box(12,44,90,58,0);
    color(0,0,3);
    box(12,60,99,74,0);
    mode = 1;
    goto wait;
}
else if(choice&J_DOWN)
{
    color(0,0,3);
    box(12,44,90,58,0);
    color(3,0,1);
    box(12,60,99,74,0);
    mode = 2;
    goto wait;
}
else if (choice&J_B)
{
    modeselect();
}
else if(choice&J_SELECT)
{
    if (mode == 1)
    {
        DIRECTION_D = 0x07U;
        DATAOUT_D = 0x02U;
        spectrum();
    }
    else if (mode == 2)
    {
        DIRECTION_D = 0x07U;
        DATAOUT_D = 0x03U;
        sweep_mode();
    }
}
else
{
    goto wait;
}
}
//***** Mode Select *****
void modeselect()
{
    UINT8 choice,mode;
    l = 0;ll = 0;lmax = 0;lloop = 0;
    clear();
    gotogxy(8,1);color(3,0,SOLID);
    gprintf("GBGSM");
    gotogxy(1,4);
    gprintf("MODE SELECT");
    gotogxy(2,6);color(2,0,SOLID);
    gprintf("SERVEY METER");
    color(3,0,1);
    box(12,44,115,58,0);
    gotogxy(2,8);color(2,0,SOLID);
    gprintf("GAMMA SPEC");
    gotogxy(0,16);
    gprintf("**Up Or Down For MODE");
    gotogxy(2,17);
    gprintf("And Then SELECT");
    mode = 1;
wait:
    choice = waitpad(J_UP | J_DOWN |
J_SELECT);
    waitpadup();
    if(choice&J_UP)
    {
        color(3,0,1);
        box(12,44,115,58,0);
        color(0,0,3);
        box(12,60,99,74,0);
        mode = 1;
    }
}

```

```

        goto wait;
    }
    else if(choice&J_DOWN)
    {
        color(0,0,3);
        box(12,44,115,58,0);
        color(3,0,1);
        box(12,60,99,74,0);
        mode = 2;
        goto wait;
    }
    else if(choice&J_SELECT)
    {
        if (mode == 1)
        {
            DIRECTION_D = 0x07U;
            DATAOUT_D = 0x01U;
            chkselect = 1;
            datasca[1] = 0;
            datasca[3] = 0;
            datasca[7] = 0;
            rate_meter();
        }
        else
        {
            mode_spec();
        }
    }
    else
    {
        goto wait;
    }
}

void main()
{
    int curx,cury;
    init();
    init_palette();
    SPRITES_8x8;
    set_sprite_data(0, 3, cursor);
    set_sprite_tile(0,0);
    curx=80;
    cury=80;
    modeselect();
}

```

ภาคผนวก ค

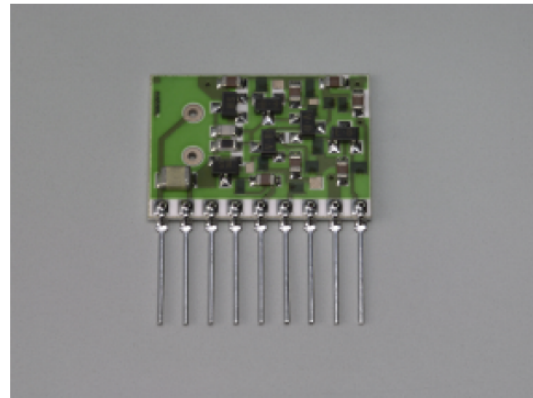
วงจรรขยายสัญญาณพัลส์ส่วนหน้า HAMAMATSU (H4083)

คุณสมบัติของวงจรมายสัญญาณพัลส์ส่วนหน้า HAMAMATSU (H4083)

6. Specifications (H4083)

Charge amplifier H4083 is a hybrid low-noise amplifier that can be used in a wide range of application, including soft X-ray and low to high energy gamma-ray spectrometers. H4083 is used in conjunction with semiconductor detectors such as Si detectors. Even when combined with a detector having a relatively large terminal capacitance, H4083 can make measurements with good energy resolution.

For optimum operation it is recommended that H4083 be used with Hamamatsu Si PIN photodiodes (S3590/S3204 series, etc.). In particular, S3590 series eliminates the risk of trouble from increased capacitance because it solders directly onto the rear side of H4083 thus reducing wire length. The increased compactness and light weight offered by this configuration make it easy to incorporate in with other test equipment, etc.



■ Specifications

Parameter	Specification
Amplification method	Charge-sensitive type
Input/output polarity	Inverted
Charge gain	0.5 V/pC
Noise characteristic	22 mV/MeV (Si)
Negative feedback constant	550 electrons/FWHM
Power supply	50 M Ω /2 pF
Power consumption	± 12 V
Configuration	150 mW
Dimensional outline	9-pin, single line type
	24 (W) \times 19 (H) \times 4 (T) mm

■ Pin connections

Pin No.	Symbol	Content
①	IN	Input terminal
②	GAIN	Feedback constant adjustment terminal
③	GND *	Input ground terminal
④	CAL	Last pulse input terminal
⑤	GND *	Power and output ground terminal
⑥	-12 V	Power terminal
⑦	+12 V	Power terminal
⑧	GND *	Output ground terminal
⑨	OUT	Output terminal
Through-hole A	P	Anode connection terminal
Through-hole B	N	Cathode connection terminal

* GND is internally connected.

The left end pin is designated No. 1 when viewed from the component side with the pins facing downwards.

● Using pin No. 2

Use pin No. 2 when changing the feedback constant which is typically 50 M Ω /2 pF. Connect feedback resistance and capacitance between pin No. 2 and No. 9. Note that this connection is made parallel to the internal feedback resistance and capacitance.

● Changing the feedback resistance and capacitance

If you want to change the feedback constant without making

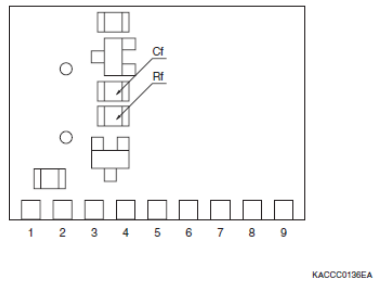
parallel connections, directly replace the resistor (Rf) and capacitor (Cf) shown at the right, which serve as feedback resistance and capacitance.

Replacement components should be 2012 size.

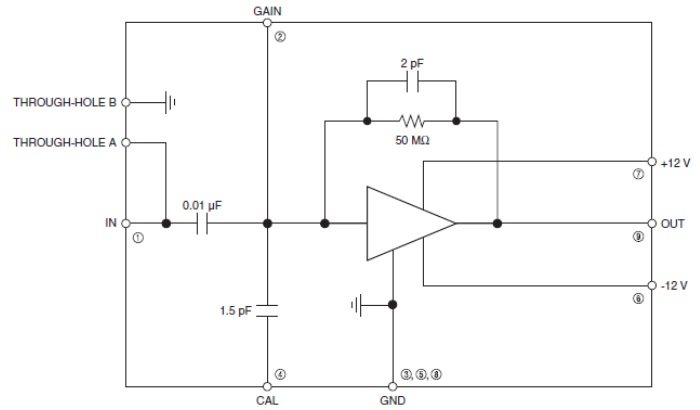
● Using through-holes A and B

When H4083 is not covered with protective coating, a photodiode (such as S3590 series) can be directly soldered on the board. In this case, insert the photodiode leads into the through-holes from the solder side.

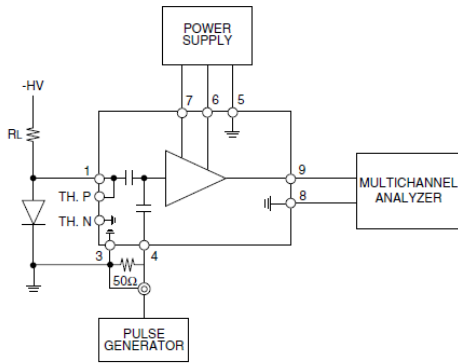
■ Component side diagram



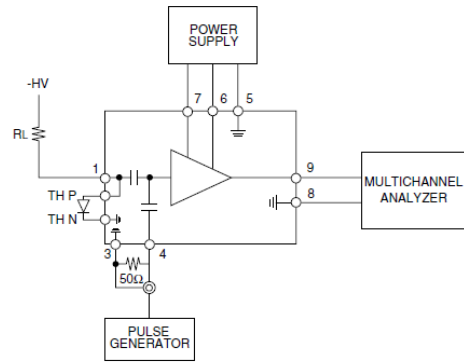
■ Equivalent circuit



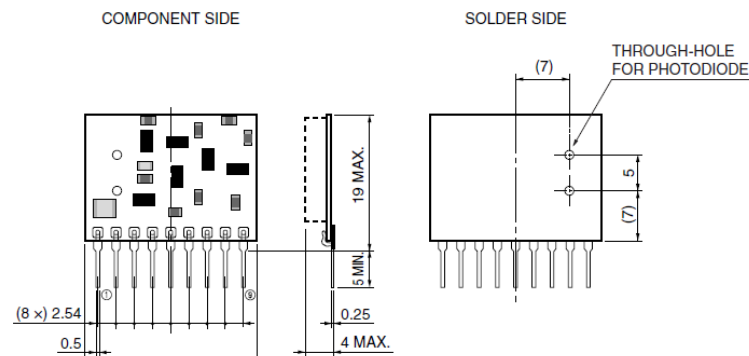
■ Circuit example 1 (external connection to photodiode)



■ Circuit example 2 (direct connection to photodiode)



■ Dimensional outline (unit: mm)



ประวัติผู้เขียนวิทยานิพนธ์

นายปิยเทพ ชอบทำกิจ เกิดวันที่ 11 ตุลาคม พ.ศ. 2527 ที่จังหวัดนครศรีธรรมราช สำเร็จการศึกษา ระดับปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชา วิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยรามคำแหง ในปีการศึกษา 2550 และได้เข้าศึกษาต่อในหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชานิวเคลียร์เทคโนโลยี ภาควิชาวิศวกรรมนิวเคลียร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2551