

บทที่ 6

โปรแกรมการทำงานของทีวีไมโครคอนโทรลเลอร์

6.1 ขอบเขตการทำงานของโปรแกรม

ทีวีไมโครคอนโทรลเลอร์จะทำงานตามโปรแกรมที่เก็บไว้ในหน่วยความจำอ่านอย่างเดียว ซึ่งหน่วยประมวลผลกลางเป็นตัวทำงานตามโปรแกรมคำสั่งที่ได้รับ โปรแกรมการทำงานนี้พัฒนาขึ้นมาจากภาษาแอสเซมบลี (Assembly Language) แล้วทำการแปลเป็นรหัสคำสั่ง (Instruction Code) ที่มีรูปแบบดังที่กล่าวในหัวข้อที่ 5.3 ด้วยตัวแปลภาษาแอสเซมบลี (Assembler) โดยมีขอบเขตการทำงานของโปรแกรม คือ

1. โปรแกรมจะสามารถควบคุมการทำงานของโทรทัศน์ ซึ่งมีหน้าที่ ดังนี้
 - 1.1 รับคำสั่งจากเครื่องควบคุมระยะไกล
 - 1.2 รับคำสั่งจากแผงควบคุมที่ตัวเครื่องรับ
 - 1.3 ปรับช่องสถานี
 - 1.4 ปรับภาพ (ความสว่าง, ความเข้ม และสี)
 - 1.5 ปรับความดัง
 - 1.6 เปิด ปิดโทรทัศน์
 - 1.7 แสดงข้อความสำหรับควบคุมบนหน้าจอโทรทัศน์
2. โปรแกรมจะสามารถถอดรหัสด้านบรรยายภาพแบบซ่อนได้ ในช่องสัญญาณ CC1 ซึ่งแสดงข้อความเป็นภาษาอังกฤษ และ CC2 ซึ่งแสดงข้อความภาษาไทย

6.2 การพัฒนาโปรแกรม

เมื่อกำหนดขอบเขตการทำงานของโปรแกรมแล้ว จึงทำการเขียนโปรแกรมโดยเริ่มจากการเขียนเป็นภาษาบรรยายโปรแกรม (Program Description Language: PDL) ซึ่งเป็นการอธิบาย

ด้วยภาษาที่เข้าใจได้ง่ายกว่าภาษาแอสเซมบลี และไม่ขึ้นกับหน่วยประมวลผลที่ใช้ ภาษาบรรยายโปรแกรมจะแบ่งออกเป็น 3 ระดับ ในแต่ละระดับจะเพิ่มรายละเอียดสำหรับเขียนโปรแกรมมากขึ้นตามลำดับ เนื่องจากโปรแกรมการทำงานของทีวีไมโครคอนโทรลเลอร์มีความซับซ้อนพอสมควร ดังนั้นในการพัฒนาโปรแกรมจึงแบ่งการเขียนโปรแกรมออกเป็น 2 ส่วน คือ โปรแกรมควบคุมโทรทัศน์ และโปรแกรมถอดรหัสคำบรรยายภาพแบบซ่อนได้ ซึ่งเป็นส่วนหนึ่งในโปรแกรมควบคุมโทรทัศน์ แล้วจึงเขียนเป็นภาษาแอสเซมบลีต่อไป

6.3 โปรแกรมควบคุมโทรทัศน์

1. ภาษาบรรยายโปรแกรมระดับที่ 1 (Program Description Language Level 1: PDL1)

MODULE: MAIN

Initialize

REPEAT

IF (key press) THEN Write code

ELSE IF (remote press) THEN Write code

ELSE Write NOP code

Read code

Code operation

IF (caption mode) THEN do CAPTION

UNTIL forever

END:

โปรแกรมเริ่มต้นทำงานจากการกำหนดเริ่มต้น (Initialize) จากนั้นจะเป็นการวนรอคำสั่งจากผู้ใช้ ถ้ามีคำสั่งใดๆ จากผู้ใช้ผ่านทางกรกดปุ่มที่แผงควบคุม (key press) หรือ เครื่องควบคุมระยะไกล (remote press) โปรแกรมจะทำการเก็บข้อมูลปุ่มที่ถูกกดเป็นรหัสในหน่วยความจำ แต่ถ้าไม่มีการกดปุ่มคำสั่งใดๆ ก็จะเป็นรหัสไม่ทำงาน (No operation: NOP) แทน จากนั้นจึงทำตามคำสั่งที่ได้รับ แล้วจึงตรวจสอบสถานะของการแสดงคำบรรยายภาพแบบซ่อนได้ หากต้องแสดงคำบรรยายภาพจะต้องไปทำงานตามโปรแกรมถอดรหัสคำบรรยายภาพต่อไป ก่อนที่จะกลับไปวนรอรับคำสั่งจากผู้ใช้ต่อไป

2. ภาษาบรรยายโปรแกรมระดับที่ 2 (Program Description Language Level 2: PDL2)

MODULE: *MAIN*

Initialize:

Clear Display Memory
 Reset Port1
 Reset Port2
 Reset All Variables (in RAM)
 Load last status of TV (bright, contrast, color and volume)
 Load last channel
 Display channel

REPEAT

IF (key press) THEN Write code

ELSE IF (remote press) THEN Write code

ELSE Write NOP code

Read code:

IF (no operation) THEN repeat

ELSE CASE (mode) IS

When (Stand By mode) =>

IF (PWR press) THEN change to TV mode

ELSE repeat

When (AV mode) =>

IF (TV/AV press) THEN change to TV mode

ELSE CASE (code) IS

When PWR => change to Stand By mode

When FUNCTION =>

IF (FUNC status = COL) THEN FUNC status = BRI

ELSE IF (FUNC status = BRI) THEN FUNC status = CONST

ELSE IF (FUNC status = CONST) THEN FUNC status = COL

When FUNCTION DOWN =>

IF (FUNC status = COL) THEN decrease COLOR

ELSE IF (FUNC status = BRI) THEN decrease BRIGHT

```

ELSE IF (FUNC status = CONST) THEN decrease CONTRAST
When FUNCTION UP =>
  IF (FUNC status = COL) THEN increase COLOR
  ELSE IF (FUNC status = BRI) THEN increase BRIGHT
  ELSE IF (FUNC status = CONST) THEN increase CONTRAST
When VOLUME DOWN => decrease VOL
When VOLUME UP => increase VOLUME
When CAPTION =>
  CASE (Caption mode) IS
    When (Off) => Change to CAPTION English mode
    When (CAPTION English) => Change to CAPTION Thai mode
    When (CAPTION Thai) => Change to Off mode
  When others => repeat
When (Preset mode) =>
  CASE (code) IS
    When PRESET => change to TV mode
    When FUNCTION DOWN => decrease VT
    When FUNCTION UP => increase VT
    When CHANNEL DOWN => decrease CH
    When CHANNEL UP => increase CH
    When MEMORY => save channel data to serial EEPROM
    When BAND => change to next BAND
    When others => repeat
When (TV mode) =>
  CASE (code) IS
    When CHANNEL DOWN => decrease CH
    When CHANNEL UP => increase CH
    When TVAV => change to AV mode
    When 0 to 9 => set CH number
    When PRESET => change to Preset mode
    When others => check other code*
  IF (caption mode) THEN do CAPTION
UNTIL forever
END:

```

ภาษาบรรยายโปรแกรมระดับที่ 2 จะกล่าวถึงรายละเอียดของโปรแกรมในส่วนของ การตั้ง
ค่าเริ่มต้น และการควบคุมโทรทัศน์ในหน้าที่ต่างๆ เพิ่มเติม

3. ภาษาบรรยายโปรแกรมระดับที่ 3 (Program Description Language Level 3: PDL3)

MODULE: MAIN

Initialize:

Clear Display Memory:

Write 00h to address F000h – FFFFh

Reset Port1:

Write 00h to P1

Reset Port2:

Write 00h to P2

Reset All Variables (in RAM):

Set all variables to initialize data

Load last status of TV (bright, contrast, color and volume):

Load Serial EEPROM

Load last channel:

Load Serial EEPROM

Display channel:

Write CH to Display Memory

REPEAT

IF (key press) THEN

IF (key1 press) THEN

Write code:

CASE (key1) IS

SYSTEM (C3h) => Write code FFh

VOL DOWN (A2h) => Write code 38h

VOL UP (81h) => Write code 39h

CH DOWN (61h) => Write code 36h

CH UP (3Fh) => Write code 37h

TV/AV (20h) => Write code 32h

Others => Write code FFh

ELSE IF (key2 press) THEN

Write code:

CASE (key2) IS

MEMORY (C3h) => Write code 40h

AUTO SEARCH (A2h)=> Write code FFh

DISPLAY (81h) => Write code 30h

PRESET (61h) => Write code 42h

FUNC DOWN (3Fh) => Write code 3Ch

FUNC UP (20h) => Write code 3Dh

FUNCTION (00h) => Write code 43h

Others => Write code FFh

ELSE IF (remote press)THEN Write code

CASE (remote data) IS

MUTE (3Fh) => Write code FFh

FUNCTION (3Eh) => Write code 3Eh

FUNC UP (3Dh) => Write code 3Dh

FUNC DOWN (3Ch) => Write code 3Ch

SYSTEM (3Ah) => Write code FFh

VOL UP (39h) => Write code 39h

VOL DOWN (38h) => Write code 38h

CH UP (37h) => Write code 37h

CH DOWN (36h) => Write code 36h

PWR (35h) => Write code 35h

SLEEP (34h) => Write code FFh

CAPTION (33h) => Write code 33h

TV/AV (32h) => Write code 32h

DISPLAY (30h) => Write code FFh

1 (2Fh) => Write code 2Fh

2 (2Eh) => Write code 2Eh

3 (2Dh) => Write code 2Dh

4 (2Ch) => Write code 2Ch

5 (2Bh) => Write code 2Bh

6 (2Ah)	=> Write code 2Ah
7 (29h)	=> Write code 29h
8 (28h)	=> Write code 28h
9 (27h)	=> Write code 27h
0 (26h)	=> Write code 26h
-- (25h)	=> Write code FFh
PP (22h)	=> Write code FFh
RET (20h)	=> Write code FFh
Others	=> Write code FFh

ELSE Write code FFh

Read code:

```

IF (code = FFh) THEN repeat
    ELSE
    CASE (mode) IS
        When (mode = Stand By) =>
            IF (code = 35h) THEN mode = TV
                                Load channel data
                                Display channel
                                ELSE repeat
        When (mode = AV) =>
            IF (code = 32h) THEN mode = TV
                                ELSE CASE (code) IS
        When (code = 35h) => mode = Stand By
        When (code = 3Eh) =>
            IF (FUNC status = COL) THEN FUNC status = BRI
            ELSE IF (FUNC status = BRI) THEN FUNC status = CONST
            ELSE IF (FUNC status = CONST) THEN FUNC status = COL
        When (code = 3Ch) =>
            IF (FUNC status = COL) THEN decrease COLOR
            ELSE IF (FUNC status = BRI) THEN decrease BRIGHT
            ELSE IF (FUNC status = CONST) THEN decrease CONTRAST
        When (code = 3Dh) =>
            IF (FUNC status = COL) THEN increase COLOR
            ELSE IF (FUNC status = BRI) THEN increase BRIGHT

```

ELSE IF (FUNC status = CONST) THEN increase CONTRAST

When (code = 38h) => decrease VOLUME

When (code = 39h) => increase VOLUME

When (code = 33h) =>

CASE (Caption mode) IS

When (mode = Off) => Mode = English
 Display Caption mode

When (mode = English) => Mode = Thai
 Display Caption mode

When (mode = Thai) => Mode = Off
 Display Caption mode

When others => repeat

When (mode = Preset) =>

CASE (code) IS

When (42h) => mode = TV

When (3Ch) => decrease VT

When (3Dh) => increase VT

When (36h) => decrease CH

When (37h) => increase CH

When (40h) => save channel data to serial EEPROM

When (32h) => IF (BAND = VL) THEN BAND = VH
 ELSE IF (BAND = VH) THEN BAND = UHF
 ELSE IF (BAND = UHF) THEN BAND = VL

When others => repeat

When (mode = TV) =>

CASE (code) IS

When (36h) => decrease CH

When (37h) => increase CH

When (32h) => mode = AV

When (26h – 2Fh) => set CH number
 Load CH data
 Display CH data

When (42h) => mode = Preset

When others => check other code*


```

IF (caption mode) THEN do CAPTION
UNTIL forever
END:

```

ในระดับที่ 3 จะมีรายละเอียดต่างๆ ของโปรแกรม และตัวแปร ก่อนที่จะนำไปรวมกับส่วนโปรแกรมถอดรหัสคำบรรยายภาพแบบซ่อนได้ แล้วเขียนเป็นภาษาแอสเซมบลีขึ้นมา

6.4 โปรแกรมถอดรหัสคำบรรยายภาพแบบซ่อนได้

1. ภาษาบรรยายโปรแกรมระดับที่ 1 (Program Description Language Level 1: PDL1)

```

MODULE: CAPTION
  Read caption data
  Decode caption data
END:

```

โปรแกรมถอดรหัสคำบรรยายภาพแบบซ่อนได้ จะอ่านข้อมูลจากรีจิสเตอร์คำบรรยายภาพ แล้วทำการถอดรหัสจากข้อมูลที่ได้รับ

2. ภาษาบรรยายโปรแกรมระดับที่ 2 (Program Description Language Level 2: PDL2)

```

MODULE: CAPTION
  Read caption data
  Load from caption register
  Decode caption data:
    IF (check parity pass) THEN
      IF (data is control code 2 times) THEN execute control code command
      ELSE IF (caption mode is English) THEN write English character
      ELSE write Thai character
    END:
END:

```

ในระดับที่ 2 ได้อธิบายรายละเอียดของถอดรหัสคำบรรยายภาพแบบซ่อนได้ ซึ่งต้องตรวจสอบข้อมูลที่รับหากเป็นรหัสควบคุม จะตรวจสอบการซ้ำกัน 2 ครั้ง จึงจะตามคำสั่งควบคุมนั้น แต่ถ้าไม่ใช่รหัสควบคุมก็จะเขียนรหัสตัวอักษรลงที่หน่วยความจำแสดงผลตามโมดการแสดงผลในขณะนั้น

3. ภาษาบรรยายโปรแกรมระดับที่ 3 (Program Description Language Level 3: PDL3)

MODULE: CAPTION

Read caption data

Load from caption register

Decode caption data

IF (check parity pass) THEN

IF (data is control code) THEN

IF (code = previous code) THEN

CASE (control code) IS

When PAC => cursor row = row defined in PAC
 cursor column = column defined in PAC
 attribute = attribute defined in PAC

When MID => attribute = attribute defined in MID

When RCL => style = pop on

When BS => cursor column = cursor column - 1
 write 00H to memory

When DER => write 00H until end of row

When RU2 => style = roll up
 caption height = 2

When RU3 => style = roll up
 caption height = 3

When RU4 => style = roll up
 caption height = 4

When FON => attribute = attribute + flash

When RDC => style = paint on

When TR => no operation

```

When RTD => no operation
When EDM => clear memory at disp. mem. of Caption Mode
When CR  => IF (style = roll up) THEN
            move data up 1 row
            scroll up
            cursor column = 1
When ENM => clear memory at non disp. mem. of Caption Mode
When EOC => swap disp. mem. and non disp. mem
When TO1 => cursor column = cursor column + 1
When TO2 => cursor column = cursor column + 2
When TO3 => cursor column = cursor column + 3
ELSE no operation
ELSE IF (caption mode is English) THEN write English character
ELSE write Thai character

```

END:

เมื่อพัฒนาภาษาบรรยายโปรแกรมทั้งสองส่วนแล้ว จึงนำมาเขียนเป็นภาษาแอสเซมบลี ความยาวประมาณ 8,400 บรรทัด, แปลเป็นรหัสคำสั่งขนาดประมาณ 9,400 ไบต์ และโปรแกรม ลงบนหน่วยความจำอ่านอย่างเดียว เพื่อใช้งานที่วิโมโครคอนโทรลเลอร์ต่อไป ซึ่งมีผลการทดสอบ การทำงานของโปรแกรมการทำงานในบทที่ 7

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย