Snake Game Simulation using LED Matrix in 8051 Microcontroller with Assembly Language

Kelompok Facebook

Mokhammad Rizqi Herdiawan	1606884893
Muhammad Sulton Tauhid	1606905885
Joshua Evans Todo	1706042850
Muhammad Hamzah	1706043065

Sistem Berbasis Komputer

Fakultas Teknik

Universitas Indonesia

2019

Deskripsi

Sistem embedded yang dibuat adalah simulasi dari game *Snake* yang dibuat dalam LED matrix dengan diatur oleh switch. Yang dilakukan adalah menggerakan ular yang ditampilkan di LED matrix 8x8. Panjang ular didefinisikan sebagai 4 LED. Gerak ular tergantung arah dari switch. Terdapat 4 switch, UP, DOWN, LEFT, RIGHT. Ketika switch UP ditekan, ular akan bergerak ke atas. Ketika switch DOWN ditekan, ular akan bergerak ke bawah. Dan sisanya sama seperti demikian.

LED Matrix 8x8 yang digunakan menggunakan 2 buah port 8-bit, port P1 dan P2. Dalam program, P1 didefinisikan sebagai variabel HORZ dan P2 sebagai VERT. Konsep penerapan LED matrix bekerja seperti koordinat dan bekerja active-low. Misalnya LED paling kiri atas akan menyala apabila bit P1.7 dan P2.7 sama-sama bernilai nol. Port P1 bekerja di garis horizontal dan port P2 bekerja di garis vertikal.

Switch yang digunakan untuk menentukan arah gerak ular menggunakan port P3. Bbit port P3.3 untuk ke atas, bit port P3.2 untuk ke bawah, bit port P3.1 untuk ke kiri, dan bit port P3.0 untuk ke bawah. Terdapat pula nilai dummy yang menggunakan port P0, dengan konfigurasi yang sama dengan switch. Begitu switch ditekan, bit nilai dummy akan di-clear sesuai switch yang ditekan. Nilai dummy ini bertujuan untuk mempertahankan gerak ular jika tidak ada perubahan arah pada ular (switch yang ditekan tidak berubah dari sebelumnya).

Dibuat data yang disimpan di alamat 120H dan 150H yang bertujuan untuk mencocokkan kondisi LED matrix sehingga mengetahui mekanisme berbelok dari ular. Data di alamat 120H mewakili kondisi posisi ular jika dilihat sebagai 4 titik. Sedangkan data di alamat 150H mewakili kondisi ular jika dilihat sebagai 1 titik. Contohnya, ketika ular berada dalam kondisi horizontal (bergerak ke kiri/kanan), dari port P1 ia akan terlihat sebagai 4 titik, sedangkan dari port P2 ia akan terlihat sebagai 1 titik.

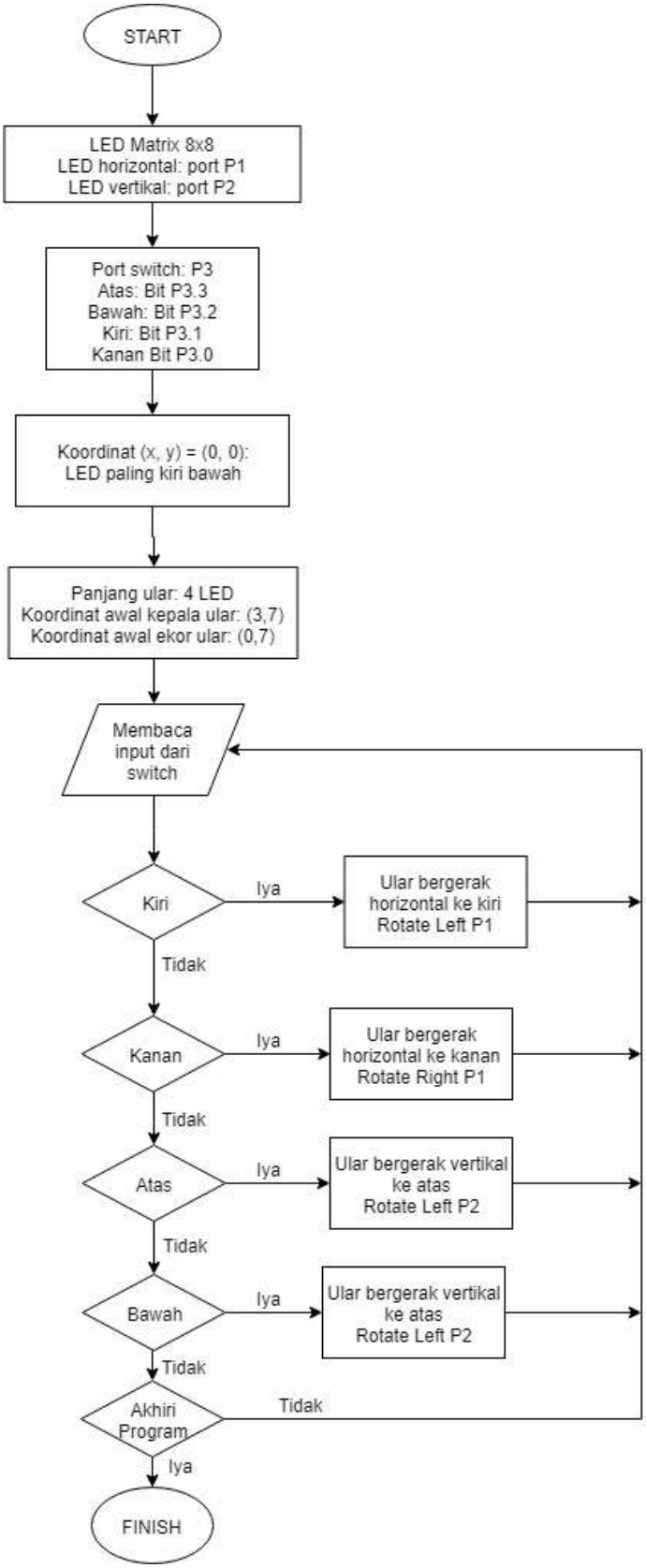
Pengondisian posisi ular dapat dapat dibagi menjadi dua bagian, yaitu kondisi vertikal dan horizontal. Ketika ular dalam kondisi vertikal, ketika geraknya atas/bawah, mekanismenya hanya perlu me-rotate nilai P2. Jika kondisi vertikal menemukan perintah gerak kanan/kiri, kondisi LED matrix akan dicocokan dengan data-data di alamt 120H dan 150H. Kemudian, ular baru akan berbelok. Lain halnya jika ular berada dalam kondisi horizontal, namun sebenarnya hanya kebalikannya saja. Pencocokan dilakukan jika terdeteksi perintah ke atas/bawah, sedangkan untuk perintah ke kanan/kiri, hanya perlu rotate nilai P1.

Pencocokan kondisi ular di LED matrix dengan data kemungkinan kondisi yang tersimpan bertujuan untuk menentukan apakah ular hendak berbelok atau tidak. Pencocokan dengan data di alamat 120H dan 150H menggunakan looping dan menggunakan indexed addressing mode. Ketika dicocokkan, ia mencocokkan dengan semua kemungkinan kondisi yang dapat terjadi di LED matrix. Begitu ular diindikasikan berbelok, akan terjadi perubahan posisi dari vertikal ke

horizontal atau sebaliknya. Jika tidak, hanya perlu dilakukan rotate saja pada port horizontal/vertikal.

Di awal program, dibuat timer delay agar dapat menentukan terlebih dahulu arah mana yang diperintahkan ke sang ‘ular’. Selain itu, timer delay juga berfungsi jika ingin membatalkan perintah untuk berbelok. Jika ular awalnya bergerakn ke kiri, kemudian ditekan tombol yang lain, cukup kembali menekan tombol kiri agar ular tidak jadi berbelok.

Flowchart


START
LED Matrix 8x8
LED horizontal: port P1
LED vertikal: port P2
Port switch: P3
Atas: Bit P3.3
Bawah: Bit P3.2
Kiri: Bit P3.1
Kanan Bit P3.0
Koordinat (x, y) = (0, 0):
LED paling kiri bawah
Panjang ular: 4 LED
Koordinat awal kepala ular: (3,7)
Koordinat awal ekor ular: (0,7)
Membaca input dari switch
Kiri
Iya
Ular bergerak horizontal ke kiri
Rotate Left P1
Tidak
Kanan
Iya
Ular bergerak horizontal ke kanan
Rotate Right P1
Tidak
Atas
Iya
Ular bergerak vertikal ke atas
Rotate Left P2
Tidak
Bawah
Iya
Ular bergerak vertikal ke atas
Rotate Left P2
Tidak
Akhiri Program
Tidak
Iya
FINISH

Source Code

```
ORG   0H
HORZ  EQU   P1    ; Baris LED untuk horizontal
VERT  EQU   P2    ; Kolom LED untuk vertikal
UP    BIT   P3.3  ; switch untuk ke atas
DOWN  BIT   P3.2  ; switch untuk ke bawah
LEFT  BIT   P3.1  ; switch untuk ke kiri
RIGHT BIT   P3.0  ; switch untuk ke kanan
DUP   BIT   P0.3  ; nilai dummy untuk kondisi ke atas
DDOWN BIT   P0.2  ; nilai dummy untuk kondisi ke bawah
DLEFT BIT   P0.1  ; nilai dummy untuk kondisi ke kiri
DRIGHT      BIT   P0.0  ; nilai dummy untuk kondisi ke kanan
MOV   P0, #0FFH   ; reset nilai dummy
MOV   HORZ, #0FH  ; reset LED horizontal
MOV   VERT, #07FH ; reset LED vertikal
JMP   MAIN        ; JUMP ke program utama

ORG   120H
DATAH:      DB    0H, 0FH, 087H, 0C3H, 0E1H, 0F0H, 078H, 03CH, 01EH, 0FH,
087H, 0C3H  ; bentuk ular dalam 4 LED
ORG   150H
DATAV:      DB    0H, 0EFH, 0F7H, 0FBH, 0FDH, 0FEH, 07FH, 0BFH, 0DFH, 0EFH,
0F7H, 0FBH  ; bentuk ular dalam 1 LED

MAIN: ACALL DELAY       ; delay agar dapat mengatur switch arah
      ; membaca switch arah
      JNB   UP, UPV
      JNB   DOWN, DOV
      JNB   LEFT, LEV
      JNB   RIGHT, RIV
      JMP   DRCT

;reset kembali nilai dummy (karena tidak pasti FFH)
UPV:
MOV   P0, #0FFH
CLR   DUP
JMP   VERTM
DOV:
MOV   P0, #0FFH
CLR   DDOWN
JMP   VERTM
LEV:
MOV   P0, #0FFH
CLR   DLEFT
JMP   HORZM
RIV:
MOV   P0, #0FFH
CLR   DRIGHT
JMP   HORZM
```

```
; menentukan arah
DRCT:
JNB   DUP, MOVU
JNB   DDOWN, MOVD
JNB   DLEFT, MOVL
JNB   DRIGHT, MOVR

; proses ular bergerak
MOVU: MOV   A, VERT
      RL    A
      MOV   VERT, A
      MOV   A, P3
      CJNE  A, #0F7H, MAIN    ; akan terus ke arah sama jika switch tidak
berubah
      JMP   MOVU
MOVD: MOV   A, VERT
      RR    A
      MOV   VERT, A
      MOV   A, P3
      CJNE  A, #0FBH, MAIN    ; akan terus ke arah sama jika switch tidak
berubah
      JMP   MOVD
MOVL: MOV   A, HORZ
      RL    A
      MOV   HORZ, A
      MOV   A, P3
      CJNE  A, #0FDH, MAIN    ; akan terus ke arah sama jika switch tidak
berubah
      JMP   MOVL
MOVR: MOV   A, HORZ
      RR    A
      MOV   HORZ, A
      MOV   A, P3
      CJNE  A, #0FEH, MAIN    ; akan terus ke arah sama jika switch tidak
berubah
      JMP   MOVR

; fungsi delay waktu dengan Timer 0 (0,05 ms)
DELAY:      MOV   TMOD, #01H
      MOV   TH0, #0FFH
      MOV   TL0, #0D2H
      CLR   TF0
      SETB  TR0
CDTMR:      JNB   TF0, CDTMR
      CLR   TR0
      CLR   TF0
      RET
```

```
; pengecekan kondisi untuk mengubah gerak ular jika akan ke arah atas/bawah
VERTM:
MOV   R2, HORZ
MOV   R3, #08H
CMPV:
MOV   DPTR, #DATAH
MOV   A, R3
MOVC  A, @A+DPTR
SUBB  A, R2
JZ    VERTN
DJNZ  R3, CMPV
JMP   DRCT

VERTN:
MOV   R2, VERT
MOV   R1, #08H
CMPY:
MOV   DPTR, #DATAV
MOV   A, R1
MOVC  A, @A+DPTR
SUBB  A, R2
JZ    MOVVR
DJNZ  R1, CMPY

; pengaturan agar ular ke posisi vertikal (ke arah atas/bawah)
MOVVR:
MOV   A, R3
MOVC  A, @A+DPTR
MOV   HORZ, A
MOV   DPTR, #DATAH
MOV   A, R1
JNB   DDOWN, SDOWN
JMP   MAKEV
SDOWN:
ADD   A, #03H
MAKEV:
MOVC  A, @A+DPTR
MOV   VERT, A
JMP   DRCT

; pengecekan kondisi untuk mengubah gerak ular jika akan ke arah atas/bawah
HORZM:
MOV   R2, VERT
MOV   R1, #08H
CMPH:
MOV   DPTR, #DATAH
MOV   A, R1
MOVC  A, @A+DPTR
SUBB  A, R2
JZ    HORZN
```

```
DJNZ  R1, CMPH
JMP   DRCT

HORZN:
MOV   R2, HORZ
MOV   R3, #08H
CMPX:
MOV   DPTR, #DATAV
MOV   A, R3
MOVC  A, @A+DPTR
SUBB  A, R2
JZ    MOVHR
DJNZ  R3, CMPX

; pengaturan agar ular ke posisi horizontal (ke arah kiri/kanan)
MOVHR:
MOV   DPTR, #DATAH
MOV   A, R3
JNB   DRIGHT, SRIGHT
JMP   MAKEH
SRIGHT:
ADD   A, #03H
MAKEH:
MOVC  A, @A+DPTR
MOV   HORZ, A
MOV   A, R1
MOV   DPTR, #DATAV
MOVC  A, @A+DPTR
MOV   VERT, A
JMP   DRCT

END
```

Link Video Program (Youtube)

https://youtu.be/4Km5oFTfTYo

Referensi

- http://www.keil.com/support/man/docs/a51/a51_st_org.htm
- https://www.elprocus.com/8051-assembly-language-programming/
- http://www.zseries.in/embedded%20lab/8051%20microcontroller/addressing%20modes%20of%208051.php#.XNiqto4zbIU
- https://www.tutorialspoint.com/embedded_systems/es_instructions.htm
- https://www.engineersgarage.com/contribution/dot-matrix-interfacing-with-8051