GPIO INTERFACING USING MSP432 (LED-SWITCH & SEVEN SEGMENT)

LAB TITLE AND CODE: EMBEDDED COMPUTING LAB (19CCE283)
EXPERIMENT NUMBER: 2
DATE: 19/04/2022 (TUESDAY)

* AIM:

To interface external peripherals, LED and Seven Segment, using MSP432 microcontroller.

* SOFTWARE REQUIRED:

Keil µVision 5 IDE (32 bit)
Publisher - ARM Ltd
Version - 5.30.0.0

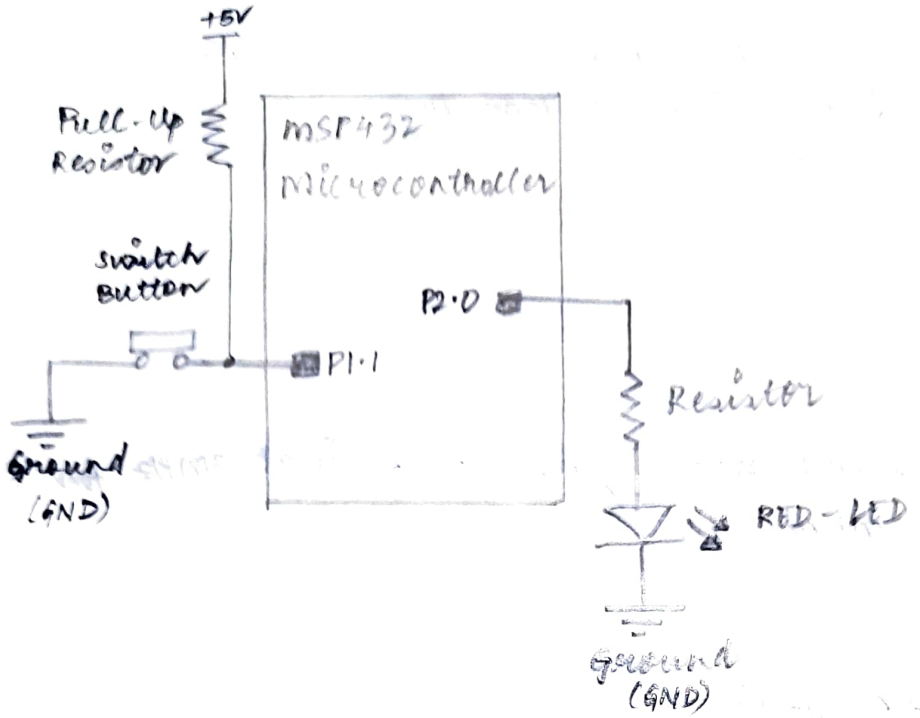
* ALGORITHM (LED-SWITCH): — (1)

(1) Configure functionality of P1.1 and P2.0 as simple GPIO Port.
(2) Configure direction of P1.1 as input pin.
(3) Enable P1.1 pull resistar; Pull up/down is selected by Px→OUT register.
(4) Configure direction of P2.0 as output pin.
(5) Use switch 1 to control the RED-LED.
(6) If not pressed, switch OFF RED-LED connected to P2.0
(7) Else, switch ON RED-LED connected to P2.0

* SOURCE CODE (LED-SWITCH):

```
/*
  p2_5.c Read a switch and write it to the LED.

  This program reads an external switch connected to P1.1 and
  writes the value to the RED-LED on P2.0. When switch is pressed,
  it connects P1.1 to ground and bit 1 of P1IN reads as '0'.

  P1.1 pin pull-up is enabled so that it is high when the switch is
  not pressed and bit 1 of P1IN reads as '1'. The LEDs are high
  active (a '1' turns ON the LED).
```

→ CIRCUIT DIAGRAM (LED-SWITCH):



Read a switch and write it to the LED.

```
  Tested with Keil 5.20 and MSP432 Device Family Pack V2.2.0 on
  XMS432P401R Rev C.
*/

#include "msp.h"

// main Function :
int main (void)
{
  P1 -> SEL1 &= ~2; // Configure P1.1 as simple GPIO Port
  P1 -> SEL0 &= ~2;

  P1 -> DIR &= ~2; // Configure direction of P1.1 as input pin
  P1 -> REN |= 2; // P1.1 pull resistor enabled.
  P1 -> OUT |= 2; // Pull up/down is selected by Px -> OUT register

  P2 -> SEL1 &= ~1; // Configure P2.0 as simple GPIO Port
  P2 -> SEL0 &= ~1;
  P2 -> DIR |= 1; // Configure direction of P2.0 as output pin

  // Infinite loop (An embedded program does not stop);
  while (1)
  {
    if (P1 -> IN & 2) // use switch 1 to control the RED-LED
      P2 -> OUT &= ~1; // If not pressed, switch OFF RED-LED
                          connected to P2.0
    else
      P2 -> OUT |= 1; // Else, switch ON RED-LED connected to P2.0
  }
}
```

PTO

* **ALGORITHM (LED - SWITCH - COUNTER) : - (2)**

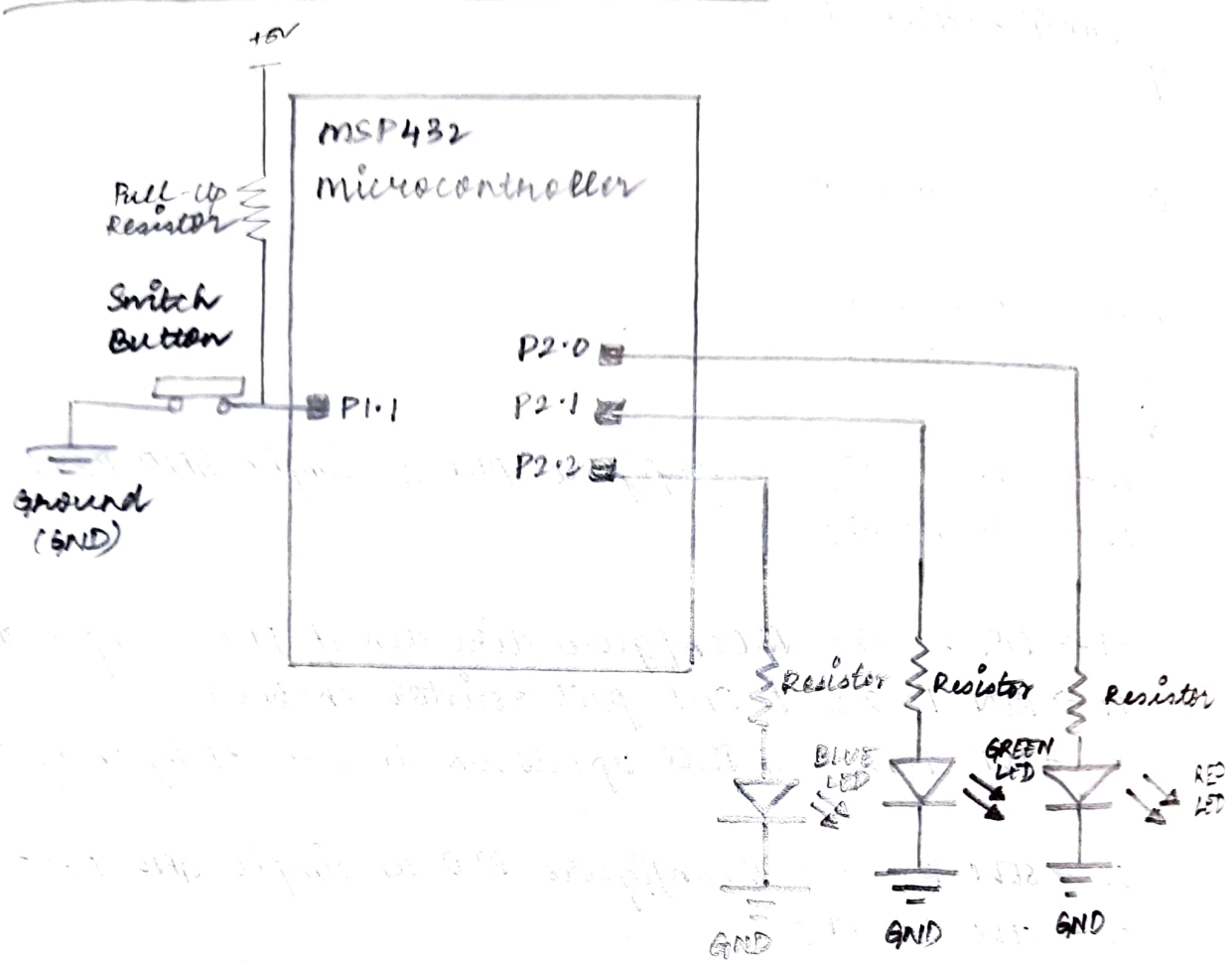
① Configure P1.1, P2.0, P2.1, P2.2 as simple GPIO Port.
② Configure direction of P1.1 as input pin.
③ Enable P1.1 pull resistor ; Pull up/down is selected by Px→OUT register.
④ Configure direction of P2.0, P2.1, P2.2 as output pin.
⑤ Use switch 1 to control the LEDs.
⑥ If pressed, switch ON the LED connected to the corresponding port.
⑦ Delay

* **CODE (LED - SWITCH - COUNTER) :**

```
#include "msp.h"
void delayMs(int n);
unsigned char i;

// main Function :
int main(void)
{
  P1->SEL1 &= ~2; // Configure functionality of P1.1 as simple GPIO port
  P1->SEL0 &= ~2;
  P1->DIR  &= ~2; // Configure direction of P1.1 as input pin

  P1->REN |= 2;  // P1.1 pull resistor enabled
  P1->OUT |= 2;  // Pull up/down is selected by Px->OUT register

  P2->SEL1 &= ~7; // Configure functionality of P2.0, P2.1, P2.2
                     (RED, GREEN and BLUE LEDs) as simple GPIO port

  P2->SEL0 &= ~7;
  P2->DIR  |= 7;  // Configure direction of P2.0, P2.1, P2.2 as
                     output pin
```

P.T.O

* CIRCUIT DIAGRAM (LED- SWITCH- COUNTER):



```
// Infinite loop (An embedded program does not stop):
while (1)
{
  if (P1 -> IN & 2) // Use switch 1 to control the LEDs
  {
    for (i = 0; i <= 7; i++)
    {
      P2 -> OUT |= i; // If pressed, switch ON the LED connected
                      to the corresponding port
      delayms (500); // Delay of 500 ms
    }
  }
}
}

// Delay milliseconds when system clock is at 3MHz for Rev C MCU:
void delayms (int n)
{
  int i, j;
  for (j=0; j<n; j++)
    for (i= 750; i>0; i--); // Delay of 1 ms
}
```

* ALGORITHM (SWITCH – SEVEN SEGMENT) : - (3)

① Define the hexadecimal values in an unsigned character array.
② Configure functionality of P1.1 as simple GPIO Port.
③ Configure direction of P1.1 as input pin.
④ Enable P1.1 pull resistor; Pull up/down is selected by Px -> OUT register.
⑤ Configure functionality of P4 as simple GPIO Port.
⑥ Configure direction of Port 4 as output pins.
⑦ Use switch 1 to control the external peripheral.
⑧ Display and Select Tens Digit of the Seven Segment.
⑨ Delay

* SEVEN SEGMENT INTERFACING (DISPLAY):

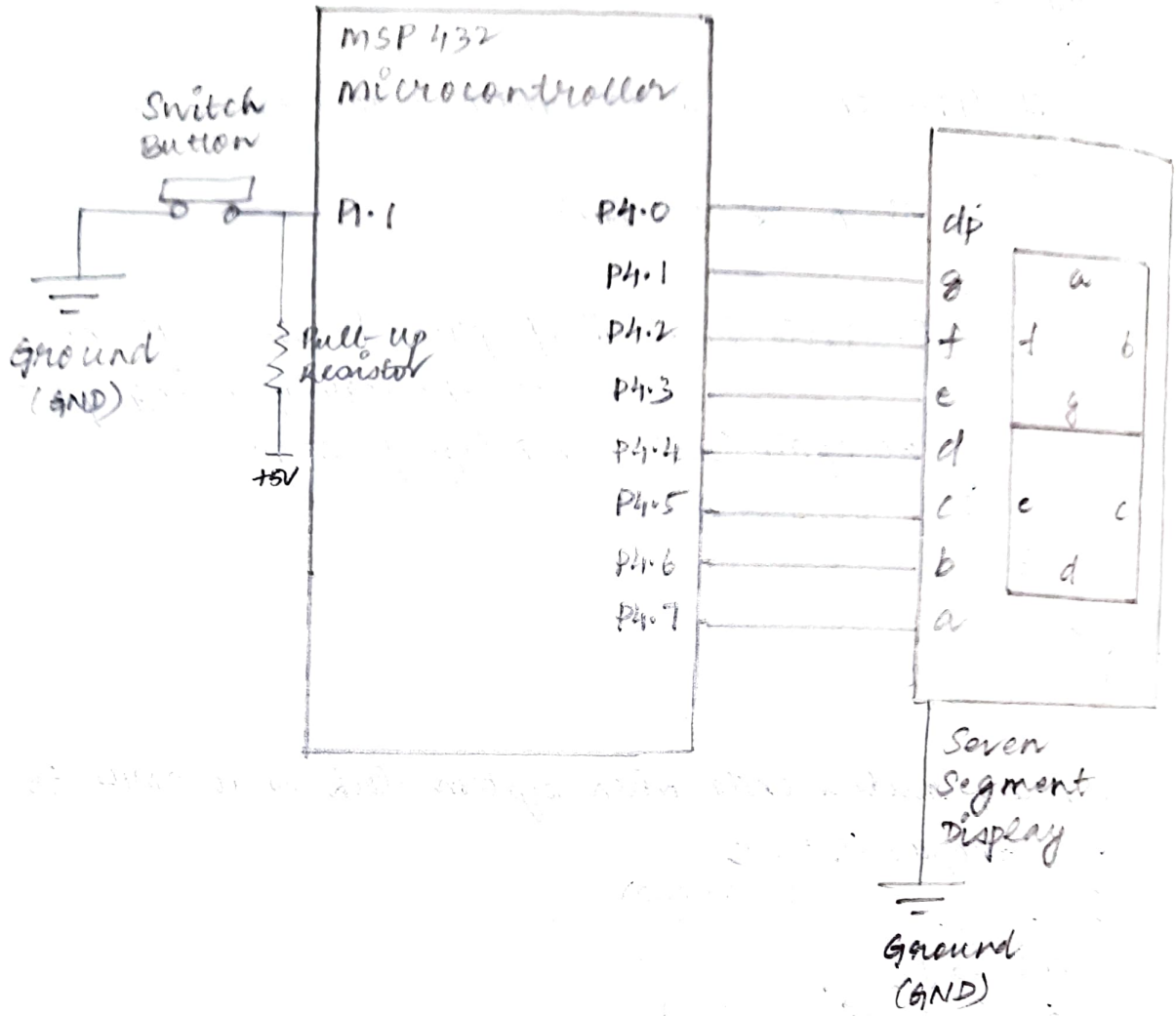
Decimal	dp (Decimal Point) P4.0	g P4.1	f P4.2	e P4.3	d P4.4	c P4.5	b P4.6	a P4.7	Hex
0	0	0	1	1	1	1	1	1	0x3F
1	0	0	0	0	0	1	1	0	0x06
2	0	1	0	1	1	0	1	1	0x5B
3	0	1	0	0	1	1	1	1	0x4F
4	0	1	1	0	0	1	1	0	0x66
5	0	1	1	0	1	1	0	1	0x6D
6	0	1	1	1	1	1	0	1	0x7D
7	0	0	0	0	0	1	1	1	0x07
8	0	1	1	1	1	1	1	1	0x7F
9	0	1	1	0	1	1	1	1	0x6F
10 (H)	0	1	1	1	0	1	1	1	0x77
11 (b)	0	1	1	1	1	1	0	0	0x7C
12 (C)	0	0	1	1	1	0	0	1	0x39
13 (d)	0	1	0	1	1	1	1	0	0x5E
14 (E)	0	1	1	1	1	0	0	1	0x79
15 (F)	0	1	1	1	0	0	0	1	0x71

Common Cathode Type

* SOURCE CODE (SWITCH - SEVEN SEGMENT):

```
#include "msp.h"
void delayMs (int n); // Delay Function
unsigned char i;

// main Function:
int main (void)
{
  const unsigned char digitPattern[] = {0x3F, 0x06, 0x5B, 0x4F, 0x66,
  0x6D, 0x7D, 0x07, 0x7F, 0x6F, 0x77, 0x7C, 0x39, 0x5E, 0x79,
  0x71}; // Hexadecimal values

  P1->SEL1 &= ~2; // Configure functionality of P1.1 as simple
                     GPIO Port
  P1->SEL0 &= ~2;
  P1->DIR &= ~2; // Configure direction of P1.1 as input pin

  P1->REN |= 2; // P1.1 pull resistor enabled
  P1->OUT |= 2; // Pull up/down is selected by Px->OUT register

  P4->SEL1 &= ~0xFF; // Configure functionality of P4 as simple
                        GPIO Port
  P4->SEL0 &= ~0xFF;
  P4->DIR |= 0xFF; // Configure direction of Port 4 as output pins

  // Infinite Loop (An embedded program does not stop):
  while (1)
  {
    if (P1->IN & 2) // Use switch 1 to control the external peripher
      for (i= 0; i< 16; i++)
      {
        P4->OUT |= digitPattern[i]; // Display Tens Digit
        P5->OUT |= 2; // Select Tens Digit
        delayMs (500); // Delay of 500 ms
      }
```

* CIRCUIT DIAGRAM (SWITCH - SEVEN SEGMENT) :



All the positive terminals (cathode) of all the 8 LEDs are connected together. All the negative terminals (anode) are left alone (ground).

```
    }
  }

// Delay milliseconds when system clock is at 8MHz for Rev C MCU:
void delayMs(int n)
{
  int i, j;
  for(j=0; j<n; j++)
    for(i=750; i>0; i--); // Delay of 1ms
}
```

* **INFERENCE:**

Interface external peripherals, LED and seven segment, using MSP32 microcontroller and all simulation results were verified successfully.