**LAB 2 Pass**

```
Define buttonstate, ledd10, ledon as boolean variables

setup()
{
    Initialize Serial communication at 9600 baud rate

    for i from 2 to 5 do
    {
        Set pin i as INPUT_PULLUP
    }

    for i from 8 to 11 do
    {
        Set pin i as OUTPUT
    }
}

loop()
{
    for j from 2 to 5 do
    {
        Read the state of button j and store it in buttonstate

        if (buttonstate is not pressed)
        {
            if (j - 1 is equal to 1)
            {
                if (ledon is equal to 0)
                {
                    Turn ON LED on pin 11
                    Delay for 500 milliseconds
                    ledon = 1
                }
                else
                {
                    Turn OFF LED on pin 11
                    Delay for 500 milliseconds
                    ledon = 0
                }
            }
        }
    }
}
```

**Lab 2 Pass Plus**

```
Define buttonstate, ledd10, ledon as boolean variables
Define tim as an integer variable

setup()
{
    Initialize Serial communication at 9600 baud rate

    for i from 2 to 5 do
    {
        Set pin i as INPUT_PULLUP
    }

    for i from 6 to 11 do
    {
        Set pin i as OUTPUT
    }

    Set tim to 1000
}

loop()
{
    Turn ON LED on pin 10
    Delay for tim milliseconds
    Turn OFF LED on pin 10
    Delay for tim milliseconds

    If button on pin 3 is pressed
    {
        Set tim to 100
    }

    If button on pin 4 is pressed
    {
        Set tim to 1000
    }

    If button on pin 5 is pressed
    {
        For i from 0 to 255 do
        {
            Set the PWM value of pin 9 to i
            Set the PWM value of pin 6 to i
            Delay for 10 milliseconds
```

```
        }

        For i from 255 to 0 do
        {
            Set the PWM value of pin 9 to i
            Set the PWM value of pin 6 to i
            Delay for 10 milliseconds
        }
    }
}
```

**Lab 2C**

```
Define pins for rows and columns
Initialize state as false

setup()
{
    // Define pins for rows and columns
    for i from 0 to 7 do
    {
        Set pinMode for rows[i] as OUTPUT
        Set pinMode for cols[i] as OUTPUT
    }
    Set pinMode for pin 4 as INPUT_PULLUP
}

loop()
{
    If button on pin 4 is not pressed
    {
        For j from 0 to 15 do
        {
            For i from 0 to 9999 do
            {
                Call disp(j)
            }
            Delay for 1000 milliseconds
        }
        Call clear()
    }
}

disp(x)
{
    For i from 0 to 7 do
```

```
    {
        Set col[i] LOW
        For j from 0 to 7 do
        {
            Set row[j] to the value in matrix[x][j][i]
        }
        For j from 0 to 7 do
        {
            Set row[j] LOW
        }
        Set col[i] HIGH
    }
}

clear()
{
    For i from 0 to 7 do
    {
        Set row[i] LOW
        Set col[i] LOW
    }
}
```

**LAB 3 Pass**

```
Define ldrpin constant as 2
Define GAMMA constant as 0.7
Define RL10 constant as 50

setup()
{
    Initialize pin 2 as INPUT_PULLUP
    Initialize Serial communication at 9600 baud rate

    for i from 3 to 10 do
    {
        Initialize pin i as OUTPUT
    }
}

loop()
{
    Read analog value from A1 and store it in analogValue

    if analogValue is greater than 15 then
    {
```

```
        Print analogValue to Serial
        for i from 3 to 10 do
        {
            Set pin i to LOW
        }
    }
    else
    {
        Call disp()
        Print analogValue to Serial
        Delay for 500 milliseconds
    }
}

disp()
{
    for i from 3 to 10 do
    {
        Set pin i to HIGH
    }
}
```

**Lab 3 pass plus**

Define ldrpin constant as 2

Define GAMMA constant as 0.7

Define RL10 constant as 50

setup()

{

Initialize pin 2 as INPUT_PULLUP

Initialize Serial communication at 9600 baud rate

for i from 3 to 11 do

{

Initialize pin i as OUTPUT

}

}

loop()

{

Read analog value from A0 and store it in analogValue

for i from 3 to 11 do

{

Set the PWM value of pin i to analogValue

}

}

**Lab 3 C**

Define ldrpin constant as 5

Define GAMMA constant as 0.7

Define RL10 constant as 50

setup()

{

Initialize pin 5 as INPUT_PULLUP

Initialize Serial communication at 9600 baud rate

Initialize pin 6 as OUTPUT

}

loop()

{

Read analog value from A2 and store it in analogValue

Set the PWM value of pin 6 to analogValue

}

**Lab 4 Pass & Pass Plus**

Define opnum as an integer

Define ledon as an integer and initialize it to 0

```
Define tim as an integer and initialize it to 200
Define once as a boolean and initialize it to true

setup()
{
   Initialize Serial communication at 9600 baud rate
   Create an integer array opt with values {1, 2, 3, 4}
   Print available options to Serial
   Initialize pin 11 as OUTPUT
   Initialize pin 9 as OUTPUT
   Initialize pin 6 as OUTPUT
}

loop()
{
   Turn ON LED on pin 10
   Delay for tim milliseconds
   Turn OFF LED on pin 10
   Delay for tim milliseconds

   If Serial data is available
   {
      Read a character from Serial and store it in ch
      If ch is a digit
      {
         Convert ch to an integer and store it in opnum
      }
   }
```

```
If opnum is equal to 1
{
   If ledon is equal to 0
   {
      Turn ON LED on pin 11
      Delay for 500 milliseconds
      Set ledon to 1
   }
   Else
   {
      Turn OFF LED on pin 11
      Delay for 500 milliseconds
      Set ledon to 0
   }
   Set opnum to 9
}

Switch on opnum
{
   Case 2:
      Set tim to 100
      Break

   Case 3:
      Set tim to 500
      Break

   Case 4:
      Loop from i = 0 to 255
```

```
        {
            Set PWM value of pin 6 to i
            Set PWM value of pin 9 to i
            Delay for 100 milliseconds
        }
    }
}
```

**Lab 4 Credit**

Include necessary libraries: Adafruit_GFX, Adafruit_ST7735, SPI, RTClib

Define RTC_DS1307 object as rtc

Define constants for TFT display pins:

- TFT_CS
- TFT_RST
- TFT_DC
- TFT_SCLK
- TFT_MOSI

Initialize Serial communication at 9600 baud rate

Initialize RTC with DS1307 module and set the initial date and time

Initialize Adafruit_ST7735 object tft with specified pins

```
setup()
{
    Initialize Serial communication
    Initialize RTC
```

Set the RTC date and time to September 21, 2023, 10:08:00

Initialize TFT display with appropriate settings

Set display rotation to 270 degrees

Set the text cursor position on the display

Clear the display (fill with black)

}

loop()

{

Read the current date and time from the RTC and store it in the "now" variable

Clear the TFT display (fill with black)

Set the text cursor position on the display

Print the current hour, minute, and second from the "now" variable on the display

Delay for 1000 milliseconds (1 second) to update the display

}

**Lab 4 Distinction**

Include necessary libraries: "lib.h" and "DFRobot_BMX160.h"

Define rows and cols arrays for the LED matrix pins

setup()

{

Initialize Serial communication at 9600 baud rate

Delay for 100 milliseconds

For i from 0 to 7 do

```
  {
    Set pinMode for rows[i] as OUTPUT
    Set pinMode for cols[i] as OUTPUT
  }

  Initialize the BMX160 sensor
  If initialization fails, print an error message and enter an infinite loop
}

loop()
{
  Declare sensor data structures for magnetometer (Omagn), gyroscope (Ogyro), and accelerometer (Oaccel)

  Read sensor data into Omagn, Ogyro, and Oaccel using bmx160.getAllData()

  Extract the accelerometer value for the x-axis into the variable x

  If x is greater than 0 and x is greater than Oaccel.y
  {
    Print "X" and the value of Oaccel.x to Serial
    Clear the LED matrix
    Call disp(0) to display pattern 0
  }
  Else if x is less than 0 and x is less than Oaccel.y
  {
    Print "Y" and the value of Oaccel.y to Serial
    Clear the LED matrix
    Call disp(3) to display pattern 3
```

```
      Delay for 500 milliseconds
   }
   Else if Oaccel.y is greater than 0 and Oaccel.y is greater than Oaccel.x
   {
      Print "Y" and the value of Oaccel.y to Serial
      Clear the LED matrix
      Call disp(2) to display pattern 2
      Delay for 500 milliseconds
   }
   Else if Oaccel.y is less than 0 and Oaccel.y is less than Oaccel.x
   {
      Print "Y" and the value of Oaccel.y to Serial
      Clear the LED matrix
      Call disp(1) to display pattern 1
      Delay for 500 milliseconds
   }
}

disp(x)
{
   Loop for k from 0 to 1999
   {
      Loop for i from 0 to 7
      {
         Set the appropriate column to LOW
         Loop for j from 0 to 7
         {
            Set the LED at row[j], column[i] to the value in matrix[x][j][i]
         }
```

```
        Loop for j from 0 to 7
        {
           Set row[j] to LOW
        }
        Set the appropriate column to HIGH
      }
   }
}

clear()
{
   Loop for i from 0 to 7
   {
      Set row[i] to LOW
      Set col[i] to LOW
   }
}
```

**Lab 5 Pass & Pass Plus**

Include necessary libraries: Adafruit_GFX, Adafruit_ST7735, Adafruit_SSD1306, SPI, avr/pgmspace

Define TFT display pins:

- TFT_CS
- TFT_RST
- TFT_DC
- TFT_SCLK
- TFT_MOSI

Create an Adafruit_ST7735 object tft with specified pins

Define an array of strings li containing PROGMEM strings

Define an array of strings buf to store the converted PROGMEM strings

Define PROGMEM strings for student ID, student name, university, full university name, semester, and year

Define a character array c_ar

setup()

{

  Initialize Serial communication at 9600 baud rate

  Initialize TFT display with appropriate settings

  Set display rotation to 270 degrees

  Fill the screen with black

  Convert PROGMEM strings to regular strings and store them in buf

  Print the first string from buf to Serial

}

String constro(const char chh[])

{

  Initialize an empty string str

  Loop through each character in the PROGMEM string chh

  {

    Read a character c from PROGMEM and add it to str

    Print the character c to Serial

  }

  Return the constructed string str

}

loop()

{

Loop through each string in buf

{

Loop to create a scrolling effect (twice the length of buf)

{

Clear the TFT display (fill with black)

Set the cursor position on the TFT display

Set text size to 2

Print the string from buf at a shifting position

Delay to control the scrolling speed

}

Delay to pause between different strings

}

}

**Lab 6 Pass**

Include necessary libraries: Adafruit_GFX, Adafruit_ST7735, Adafruit_SSD1306, SPI, avr/pgmspace

Define TFT display pins:

- TFT_CS
- TFT_RST
- TFT_DC
- TFT_SCLK
- TFT_MOSI

Create an Adafruit_ST7735 object tft with specified pins

```
Define arrays for rows and columns for the LED matrix pins

Define an integer variable x

setup()
{
   Initialize TFT display with appropriate settings
   Fill the screen with black
   Initialize Serial communication at 9600 baud rate

   For j from 0 to 7 do
   {
      Set pinMode for rows[j] as OUTPUT
      Set pinMode for cols[j] as OUTPUT
   }

   Set pinMode for pin 2 as INPUT_PULLUP
   Attach an external interrupt to pin 2, which calls the displaymic() function on LOW signal
}

//ISR - External Interrupt
void displaymic()
{
   Set the cursor position on the TFT display
   Set text size to 5
   Print "@" on the TFT display
   Delay for 100 milliseconds
}
```

void loop()

{

// The loop is empty because the program mainly responds to the external interrupt

}

**Lab 6 Pass Plus**

Include necessary libraries: lib.h, Adafruit_GFX, Adafruit_ST7735, SPI

Define arrays for rows and columns for the LED matrix pins

Define an integer variable x and initialize it to 0

setup()

{

Initialize Serial communication at 9600 baud rate

For j from 0 to 7 do

{

Set pinMode for rows[j] as OUTPUT

Set pinMode for cols[j] as OUTPUT

}

Set pinMode for pin 2 as INPUT_PULLUP

Attach an external interrupt to pin 2, which calls the displaymic() function on LOW signal

}

// ISR - External Interrupt

void displaymic()

{

Loop for k from 0 to 9999

```
    {
      Loop for i from 0 to 7
      {
        Set the appropriate column to LOW
        Loop for j from 0 to 7
        {
          Set the LED at row[j], column[i] to the value in matrix[x][j][i]
        }
        Loop for j from 0 to 7
        {
          Set row[j] to LOW
        }
        Set the appropriate column to HIGH
      }
    }

    Increment x by 1
    Call the clear() function to turn off all LEDs
}

// Clear all LEDs
void clear()
{
    Loop for i from 0 to 7
    {
      Set row[i] to LOW
      Set col[i] to LOW
    }
}
```

```
void loop()
{
   // The loop is empty because the LED patterns are displayed in the interrupt service routine (ISR)
}
```