Final Project

General Info →

- SDI-12 Sensor & data logger
  - BME680 - Temp, Humidity etc. <Priority>
  - BH1750 Light Sensor
- sensors connected to arduino via I2C, as soon as data is recieved, put togeter & transperred to the UART interface using SDI commands

$$\begin{tikzcd}\boxed{\text{Sensors}} \arrow[r, leftrightarrow, "I_2C"] & \boxed{\text{Arduino}} \arrow[r, leftrightarrow, "\text{UART}"] & \boxed{\text{UART to SDI-12 converter}} \arrow[r, leftrightarrow, "\text{SDI-12}"] & \boxed{\text{SDI-12 to USB converter}} \arrow[r, leftrightarrow, "\text{USB}"] & \boxed{\text{computer (SDI-12 command)}}\end{tikzcd}$$

Pass : SDI-12 Sensor

- program Arduino to read relevant data from sensor
  Address 1 & 0 for sensors by default
- program Arduino vased on SDI-12 communication protocol
  > Understand at least 5 Commands

- Address Query

  → ?! , response : 0

  ∴ address = 0

- save data from sensor into address
  - how to save changing data value into single address
  - convert data into bytes

- Change Address

  → aAv!

  eg. 0A1! ∴ New Address = 1

  - does it automatically detect seconds & parameter

- need to create ~~values~~ variables for a & v
- check if address is full, if is don't overwrite

- Start Measurement

  → aM! $\quad$ OM! $\overbrace{000 3}^{3s}$ 5 → with 5 parameters

  address of SDI12 eg. OM! ∴ 00035 , 3 seconds with 5 parameters

  - if only BME 4 parameters

  ours will ve instant

- tells level logger to take a measurement
  - returns time & number of measurements

  eg. OM! 00102 time it takes for measurement

  ↑ address

- make Arduino understand
  the SDI-12 commands

• Send Data ~~→~~ , page no. (address of sensors)

→ aD0! , address of SDI12

+ gets group of data from Level Logger
  - used after M or C command (reads data & displays)

eg. 0D0! 0 + temp + pressure + humid + gas & 0 + light
  - whatever current measurement is

• Continous Measurement

→ aR0! ... aR9!

+ same as Send Data just create loop
  - can use timer interrupt

Summary

- 5 necessary comands all work together
- need to get Arduino to understand the SDI-12

+ set Address for SDI-12 sensor
  - and Address for each sensor
+ in SDI-12 need to read string (?, A, M, etc)
  - test it is in the address

?!

- get address , ? by default is 0
  - basic, give address to sensor
  - can scan for ~~empty~~ ~~a~~ occupied addresses

+ issues going from SDI-12 to Arduino
  SDI-12 , high to recieve data
  low to send data

PseudoCode →

```
include liuraries
setup SDI-12 and sensors
device address
sensor address
variaules used in functions
setup f
void setup()
    ~~if seri~~ activate serial monitor
void loop()
    have switch case for the functions
        wait for user input & it will execute
void addQuery()
    serial.println (Sensor 1 ~~data~~:
```

- can't test anything!!
    - so can't see what is wrong or right

Ruuric

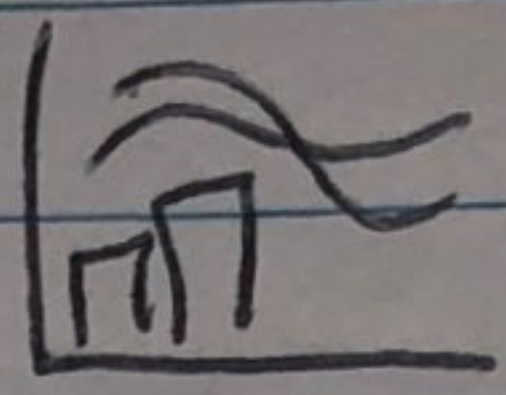
Demo (35%)

- design & implement emuedded microcontroller (20%)
    - understand design requirements
        - propose alternatives (design soln's & constraints)
        - clear evaluation & justification
- programming skills (15%)
    - efficient, logical ~~errors~~ design
        - clear explanation
- required specification (35%)
    - vs performance index
- oral communication & answering questions (20%)
    - prepared, clear
        - demonstrate knowledge
        - answer q's, in detail with elauoration
- safely execute experiments

credit

Credit : Create Menu in LCD with buttons

- can either → have 2 buttons represent each sensor
  or have button to scroll through options than
  select button
- display on LCD
  - wave or bar graph



, can only have black & white so will need option to go through sensor data

SD Card

- setup() , create new file with SD.open()
  FILE_WRITE → enables read & write
  file.println() write string to card
  SD.read() , SD. close()
  ↳ could save data into

```
myFile = SD.open ("test.txt", FILE_WRITE)
include <SPI.h> , <SD.h>
string Data . . . . .
```

- create function just for saving data to SD

Push Button Menu

- debounce buttons
  - create an array for menu options
    - button for up, down & select
      - how to make back button ?

menu options ~~arr~~ [0 . . . 4]
~~back up~~ if up = '1'
array position - 1

<Breaking Code Down>

~~include libraries~~
could have dot to show which is selected
~~setup sensors~~

- have menu in LCD (SSD1306)
    - easy enough
- start by having menu in serial.monitor

~~[illegible]~~ - pushButtons
    - deBounced?
        - for full marks, yes!

- create array for menu options
    - ~~and~~ array sensors ['Temp', 'Light' etc]
        - when option selected display data on graph

```
string menuOption[] = {{"Temp"}, {"Light"} etc}
const int buttonpins
        down, enter, clear
        int downPressCount = 0; //change menu option
buttonState
currentButtonState = Low  //for each button
lastButtonState
debounceDelay = 50
lastDebounceTime = 0;
activate LCD
setup() {
    turn on & clear LCD
    set pins to input, delay }                can add >
void loop() {                                 ~~to make easi~~
                                              &
    lcd.cursor  , print (menuOption[directionPush])
// wait for buttonPress
    current State = digitalRead (~~but~~ pin)
```

Temp > ⟶ Pressure ~~BM~~

> Temp   > + ~

create static menu : Pressure
Humidity
Gas

```
- make variable for > position
      - when ~~set~~ down pressed
            x = position ++
      ~~when select pressed~~
  for (int i=1 ; i<=4 ; i++) {
        display.setCursor (0, i)
        display.print (i == menuCount ? ">" : " ");

  menuCount ++                              when down pressed
     if menuCount > 5                       add to count
           menuCount = 1 ;
  menuOption [] = {{1}, {2} etc
  directionPush = 0 ;  ⟶  prints position
 void loop()
        display (menuOption [directionPush]) ;
- we want to print a menu, only change the position
 of the ">"
        if buttonStateDown pressed
               directionPush ++
                   // adds to count, changing position in arr
 set cursor position to ~~(0, 0)~~ 0
        &       set.cursor = (0, cursorPos)
 ~~when~~       println ~~n~~ (">")
                print (menuOption [])
- change To printMenu
                Temp            >
                Pressure

 if select press && cursor
```

```
#include "SD.h"
 File dataLogger ;
String Temp, Pressure  etc.
initialize the SD card  setup() ;
 if (!SD.vegin(10)) {
          print (failed)
          while (1) }  print (done)
 ~~Se print my~~ dataLogger = SD.open("data.txt", FILE_WRITE
 if (~~myFile~~ datalog) {
        ~~my~~dL.print ("Temp (°C), Pressure (Pa) et);
        dL.close
      } else { print (error opening
void loop () {
        if (cursor == 0)                  etc
              logSensorData(4veme.temp);

void data-logging () {
      ~~String~~ dataFile = SD.open ("dataloy.txt", FILE_W)
   if (datafile) {
           print (dataType)
                (":")
           print (value)
             close
        serial.print ("sensor logged to SD card");
```