For GitHub

ReadMe:

Project description:

Light Source and Objects Proximity Detector System – the project structured as a finite state machine. The system comprises several states:

1. **Sleep Mode (state 0):** the system is idle and conserving energy.
2. **Objects Detector (state1):** the system scans a 180-degree range, identifies objects presence, and display its findings to the user.
3. **Precise Objects check (state2):** The system moves to a specific angle, verifies object presence, and display its finding to the user.
4. **Light Source Detection (state3):** A scan identifies light source and showcases their locations to the user.
5. **Light sources and Objects Detection (state4):** a scan identifies both light sources and objects, displaying the results to the user.
6. **Script Execution (state5):** The system enters script mode, allowing programmed tasks to be stored in the MCU flash memory and to be executed by the MCU.

Each state is triggered by the user.

Through this design, the system can effectively detect objects and light sources while allowing for customizable scripting.

For this project we used MSP430FG4619 microcontroller, the code was written in C language in CCS (Code Composer Studio) IDE. For the PC side of this project we used Python and GUI for the user's interface.

Libraries included:

MCU SIDE:

- "stdio.h "
- "string.h"
- "stdint.h"
- "msp430FG4619.h"

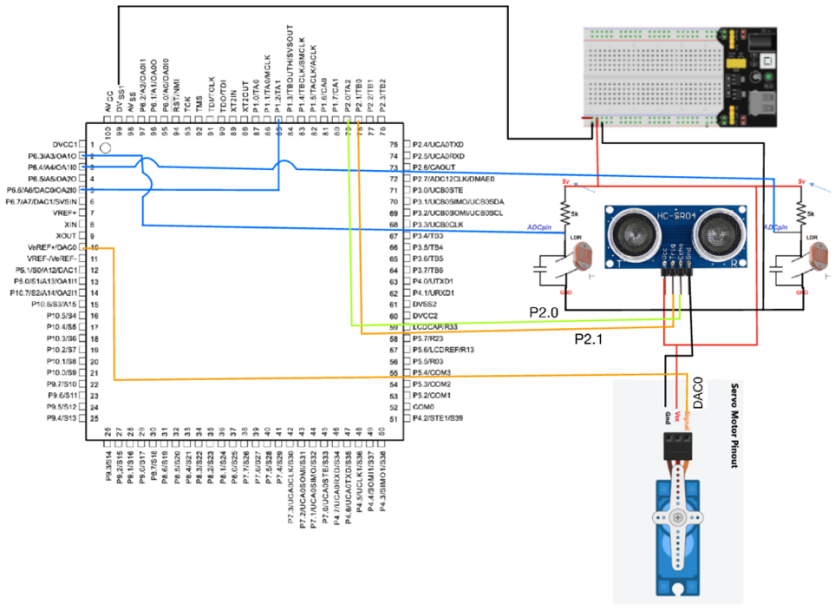
PC SIDE:

- **"serial"**
- **"time"**
- **"PySimpleGUI"**
- **"numpy"**
- **"matplotlib.pyplot"**
- **"os"**

Packages:

- **Pyserial**
- **Pysimplegui**
- **Pyautogui**
- **Numpy**
- **Matplotlib**

Hardware connections:



LCD:

Data0 – Data3 -> P1.4 – P1.7

LCD control:

E -> P2.5

RS -> P2.6

RW -> P2.7

Pushbutton:

PB2 -> P1.0 – input

Ultrasonic:

Trigger leg -> P2.1

Echo leg -> P2.0

GND -> DC voltage supplier GND

VCC -> DC voltage supplier VCC

Servo:

Signal leg -> DAC0 leg in MSP430FG4619

GND -> DC voltage supplier GND

VCC -> DC voltage supplier VCC

LDR:

LDR1 (ADC sample) -> P6.3 (channel A3 in the MCU)

LDR2 (ADC sample) -> P6.4 (channel A4 in the MCU)

DC voltage supplier:

GND -> MCU GND.

Explanations:

Timer:

The MSP430FG4619 MCU comes with two Timers, TimerA and TimerB, which can be used to take time measurements. The registers TAR for TimerA and TBR for TimerB gives the Timer's current count.
There are two modes that the Timers can be in: Compare Mode and Capture Mode, Moreover, TimerB have seven Capture\Compare registers and TimerA have three Capture\Compare registers, which gives the Timers the capabilities to support multiple Capture\Compare, PWM outputs and interval timing. Also, the Timers has extensive interrupt capabilities.

Capture Mode:

Capture Mode is used to record time events. It can be used for speed computations or time measurements.

Compare Mode:

Compare Mode is used to generate PWM signals or interrupts at specific time intervals.

UART:

The MSP430FG4619 comes with the universal synchronous\asynchronous receive\transmit (USART) peripheral interface that support two serial modes with one hardware module.
The UART module, which we use in this project, in the MSP430FG4619 MCU allows for asynchronous serial communication, which means data is transmitted without a common clock signal between the sender and the receiver. This is a particularly useful for interconnecting devices that might operate at different clock speeds or in scenarios where precise synchronization is not required.
Our UART configuration for this project includes:

- Baud rate: 9600 BPS (Bits Per Second)
- Data Size: 8 – bits
- 1 start bit
- 1 stop bit
- No Parity bit.

HC-SR04 Ultrasonic sensor:

The HC-SR04 ultrasonic sensor is a widely used distance measuring module. It operates by applying at least 10microseconds HIGH signals at the Trigger leg, it emits eight 40kHz pulses that hit the object and bounce back to the sensor, the Echo pin emits a HIGH signal that his width indicates the time between the pulses were sent and the when the reflected pulses are detected. This time delay is used to calculate the distance between the sensor and the object.
Distance Calculation:

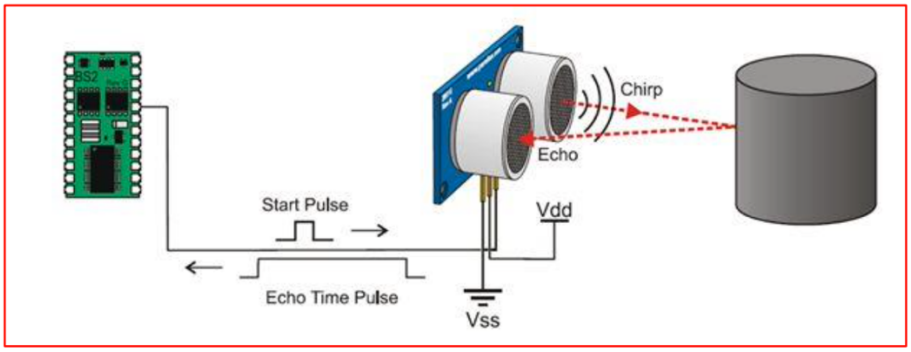
$$Range[cm] \cong EchoPulseTime \times \frac{34{,}000\left[\frac{cm}{sec}\right]}{2}$$
$$= EchoPulseTime \times 17{,}000\left[\frac{cm}{sec}\right], when\ 34{,}000\left[\frac{cm}{sec}\right] is\ the\ speed\ of\ sound\ c.$$

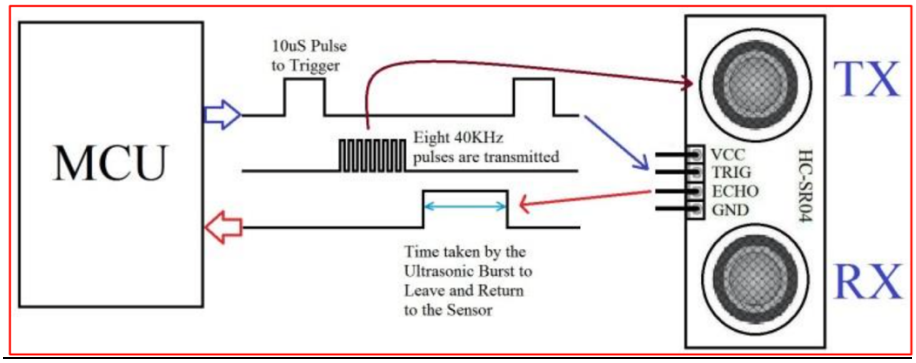
To be more precise, the speed of sound is depended in the temperature, assuming we are inside a lab and the temperature is 25 Celsius then

$$c\left[\frac{m}{sec}\right] = 331.3 + 0.606 \times TemperatueInCelsius = 346.45\left[\frac{m}{sec}\right] = 34{,}645\left[\frac{cm}{sec}\right]$$
$$Range[cm] = EchoPulseTime \times 17{,}322.5$$

In order to calculate the Echo Pulse Time, we used the Input Capture mode in TimerA at layer 3, we configured this register to raise the interrupt flag when there is a raising or failing edge. In the interrupt vector we placed the value that was captured in the TimerA inside two variables, one variable for falling edge and the other for raising edge. By subtract one variable from the other we know how many MCU clocks were between the raising edge and the falling edge, TimerA counts in $2^{17}[Hz]$ so the Echo Pulse Time will be:

$$EchoPulseTime = Number\ of\ clock \times \frac{1}{2^{17}}$$





Servo Motor:

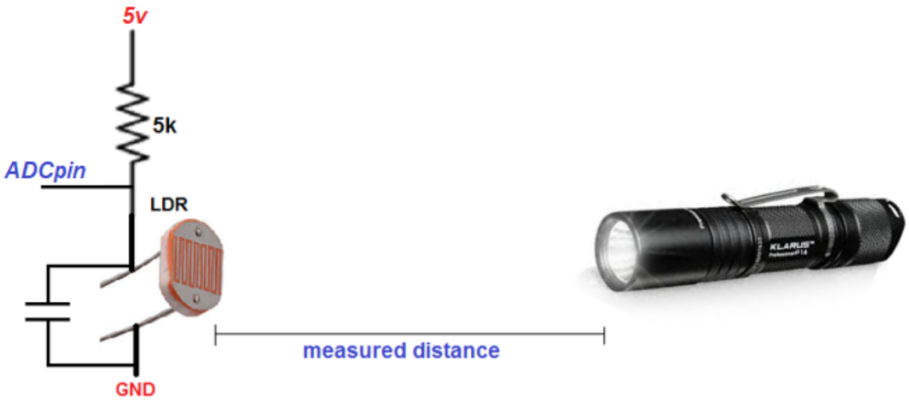
A servo motor is a small electromechanical device utilized for controlling precise angular positions. It responds to input PWM signals (Max frequency = 50[Hz]), and adjusts its shaft to a designated angle. By adjusting the duration of the High-level signal within the PWM, the motor's shaft is positioned at a specific angle. For out project's 180-degree scan, a PWM signal at 50Hz with a High-level time of 0.48msec corresponds to the angle 0 degrees, while a High-level of 2.1msec achieves 180 degrees. The relationship between angle "R" and High-level time is given by the linear transformation:

$$T_{on}(R) = 0.0089 \times R + 0.48$$

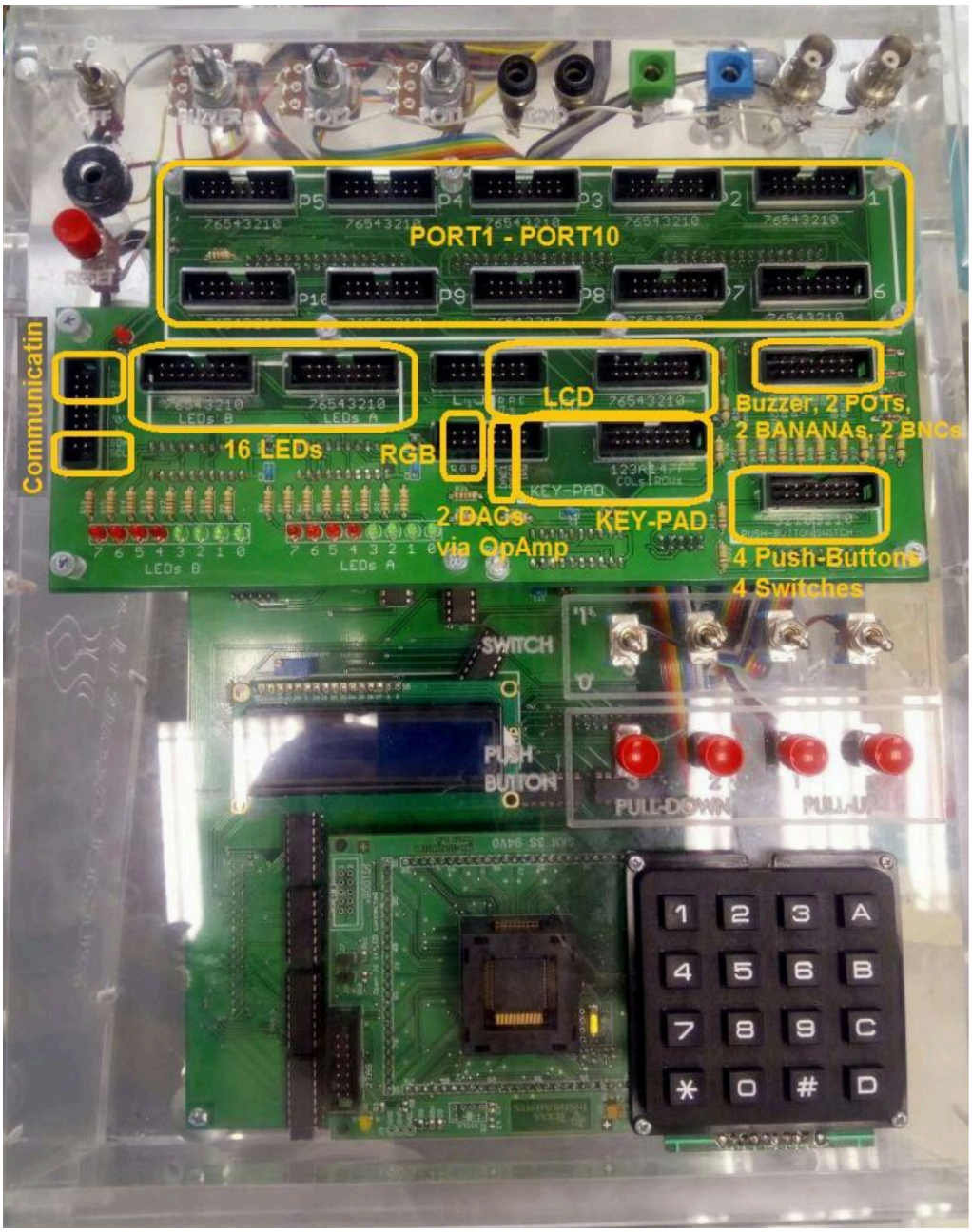
LDR:

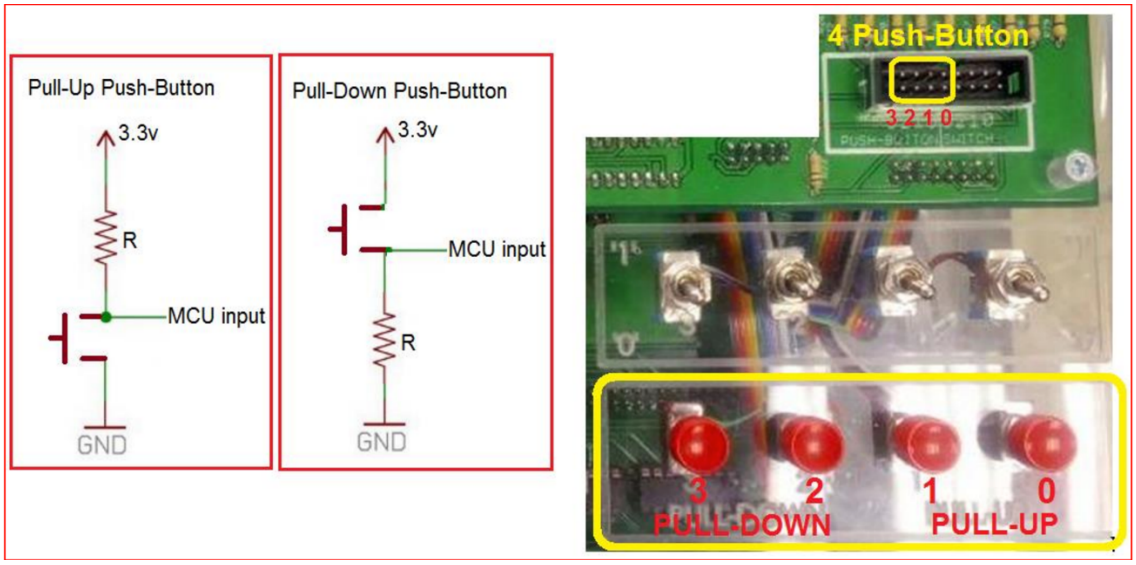
An LDR, or Light Dependent Resistor, is a type of resistor whose resistance varies based on the intensity of light it is exposed to. As light increases, the resistance of the LDR decreases, and vice versa. In our project we used two LDR, placed them on each side of the ultrasonic sensor and used them to detect light sources in the area. In the MSP430FG4619 there is a 12-bit ADC peripheral module, we use him to sample the voltage that drops on the LDR, as the resistance decreases so as the voltage, and from the voltage sampled we can know the distance of the light source from the LDRs.

In order to determine the specific distance of the light source we first need to calibrate the LDRs data to the specific light source and specific area of the scan.

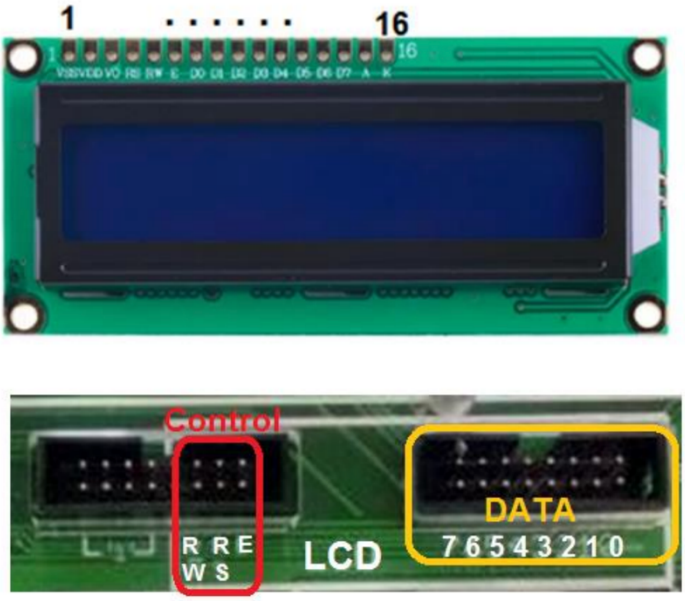

5v
5k
ADCpin
LDR
GND
measured distance

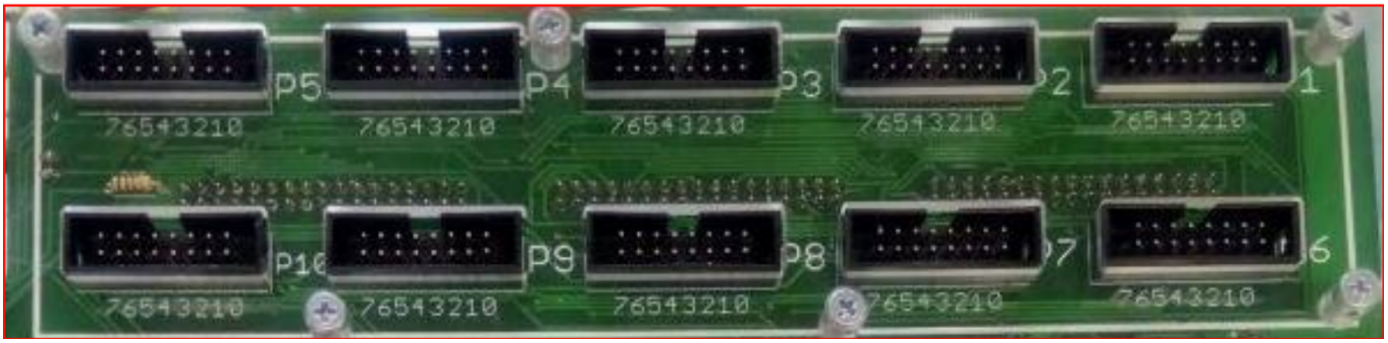
The MSP430FG4619 we used :

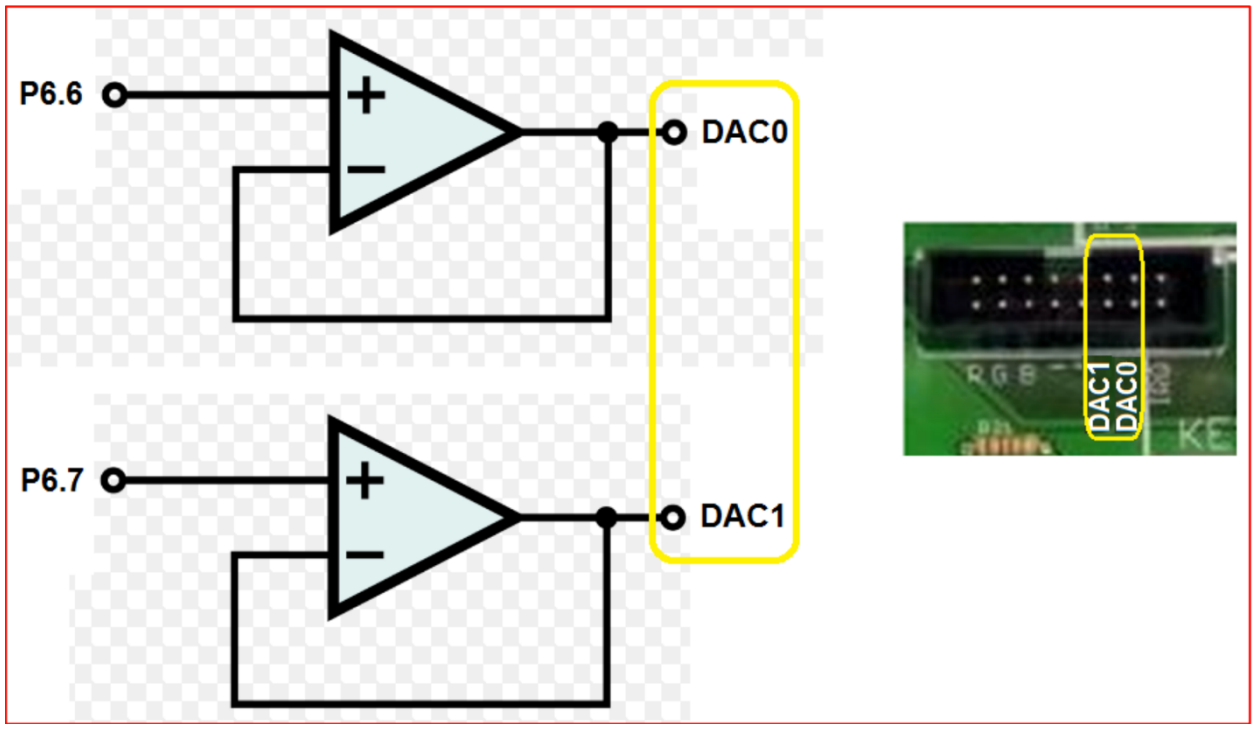




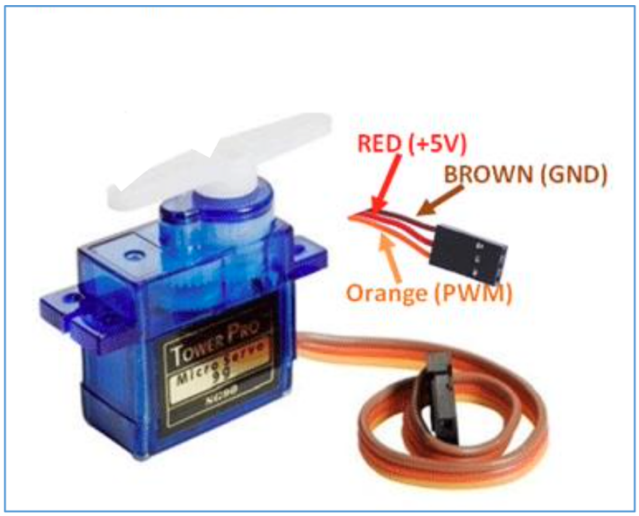
PIN	Symbol	Function
1	VSS	GND
2	VCC	+5V
3	VEE	Contrast
4	RS	data/instruction
5	R/W	read/write
6	E	Enable
7	D0	data bus
8	D1	data bus
9	D2	data bus
10	D3	data bus
11	D4	data bus
12	D5	data bus
13	D6	data bus
14	D7	data bus
15	Anode	LED (+)
16	Kathode	LED (-)

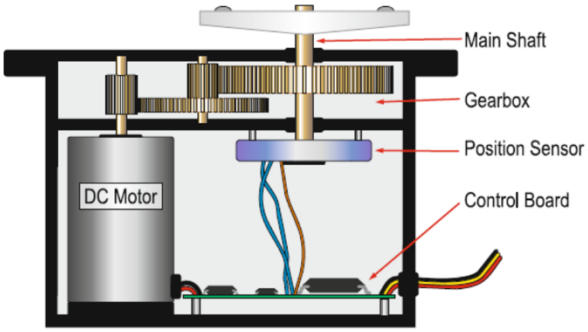


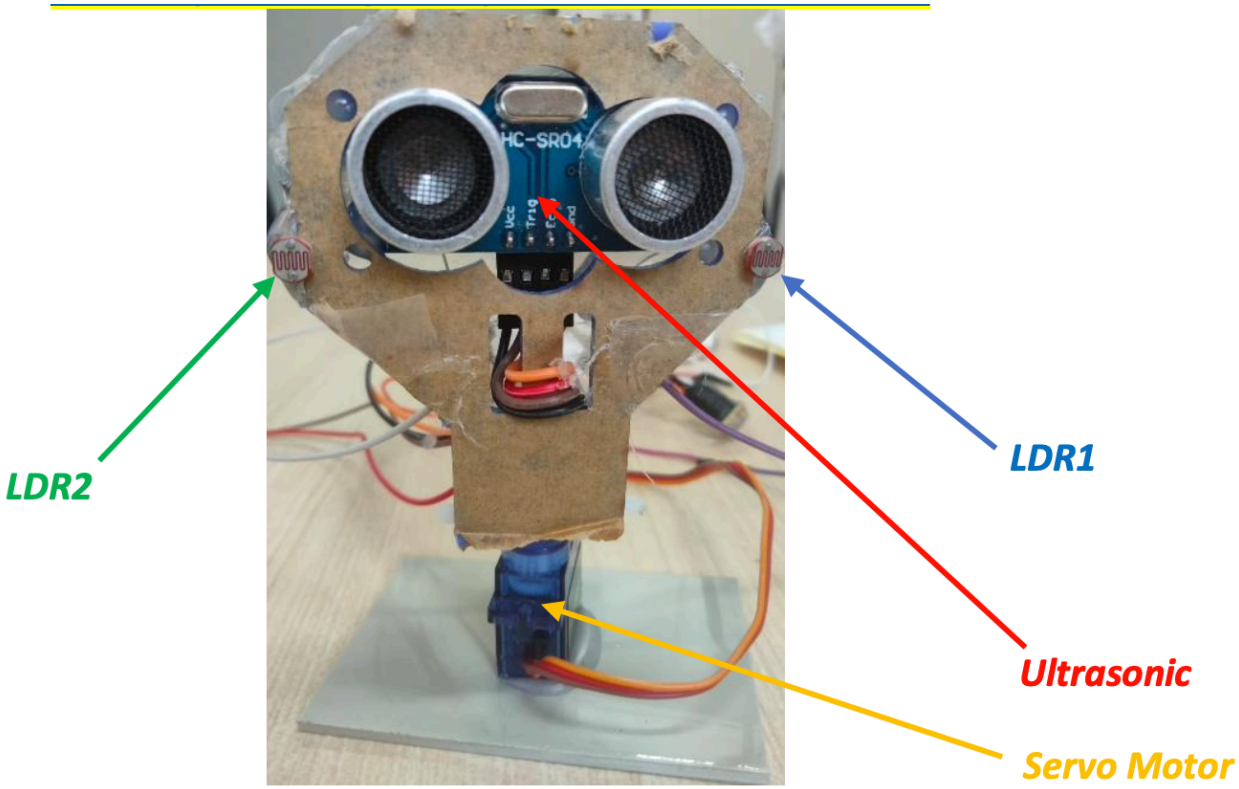



P6.6
DAC0
P6.7
DAC1
DAC1
DAC0

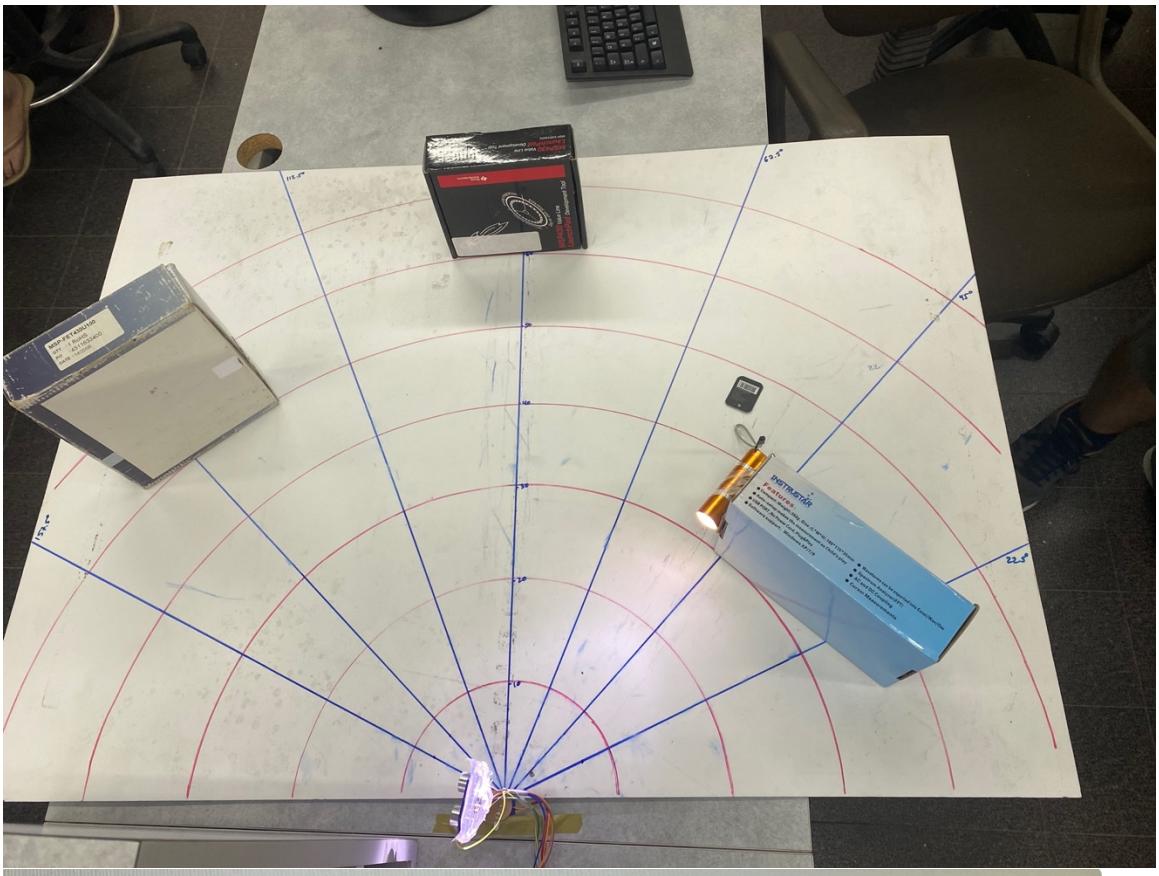
The devices:

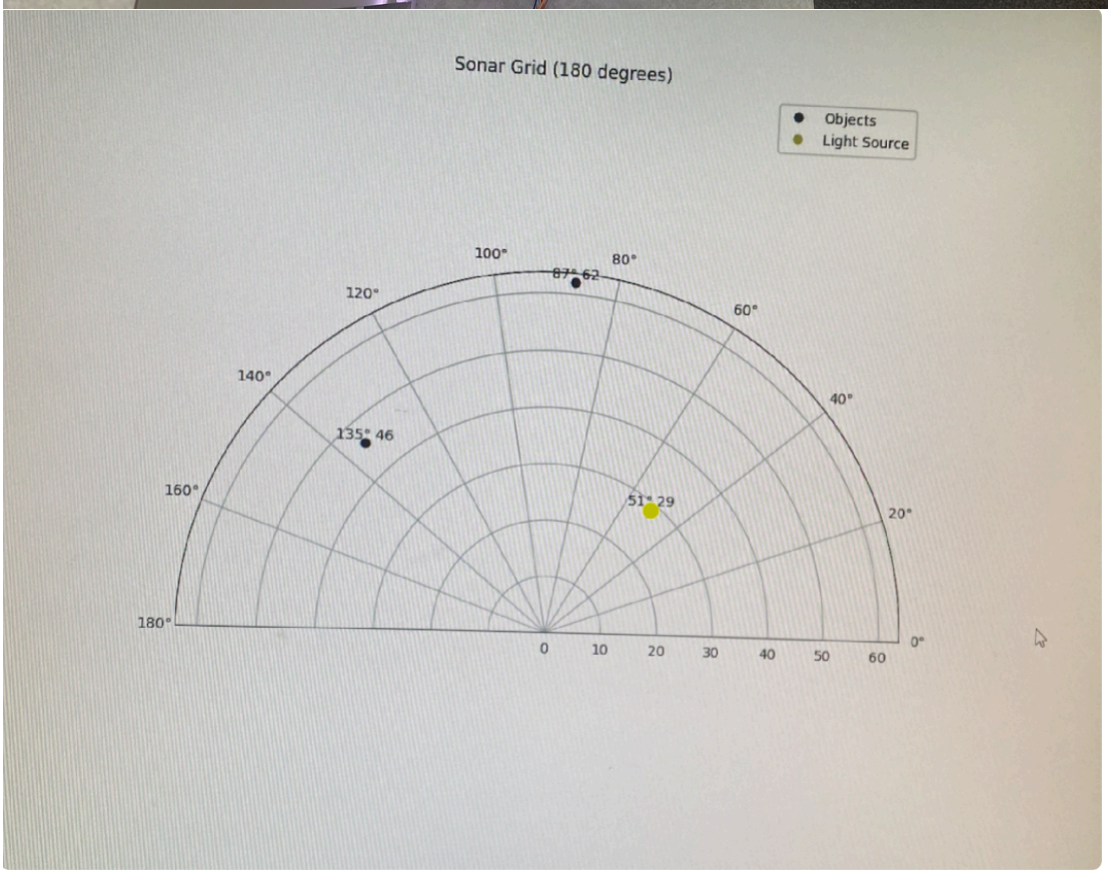

RED (+5V)
BROWN (GND)
Orange (PWM)

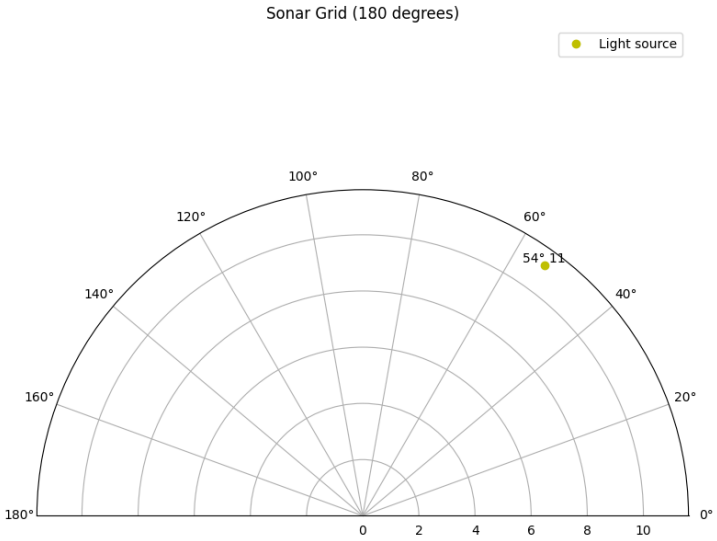

Main Shaft
Gearbox
Position Sensor
Control Board
DC Motor

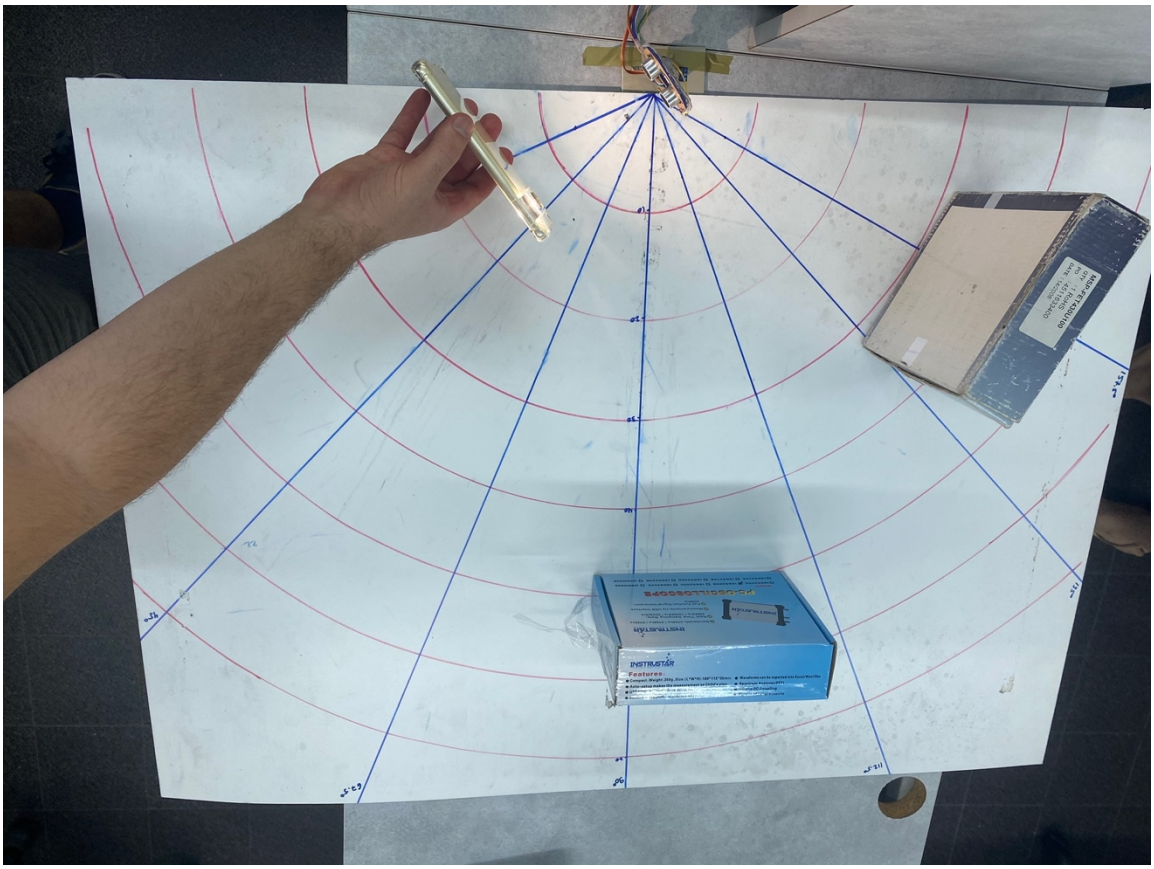


Examples:

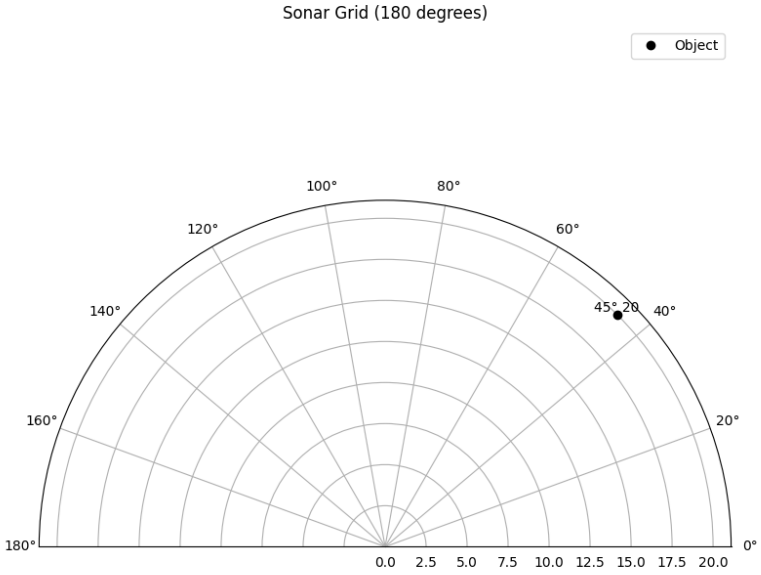



Sonar Grid (180 degrees)
Objects
Light Source
100°
80°
120°
60°
140°
40°
160°
20°
180°
0°
87° 62
135° 46
51° 29
0
10
20
30
40
50
60

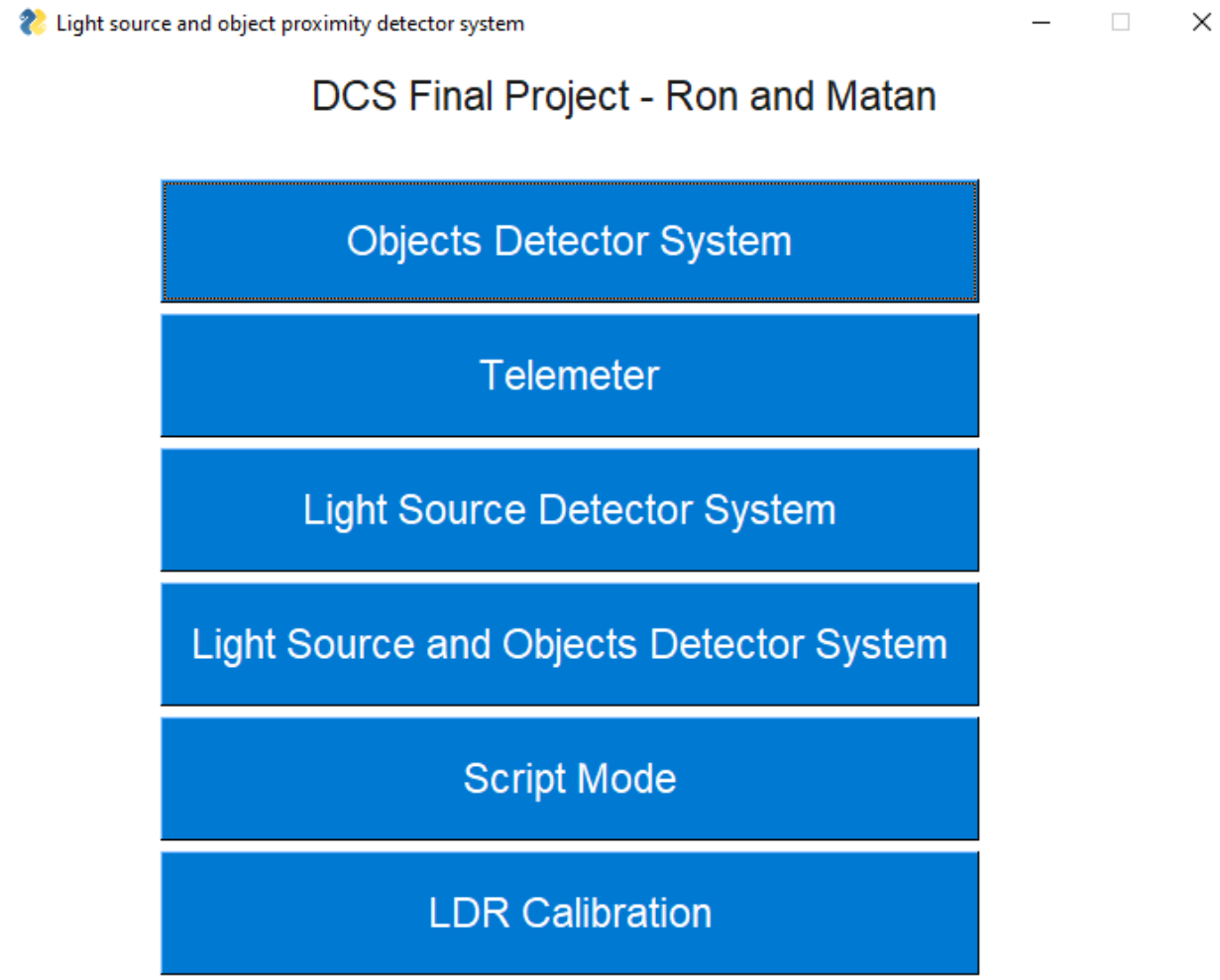

Sonar Grid (180 degrees)
Light source
100°
80°
120°
60°
54° 11
140°
40°
160°
20°
180°
0°
0
2
4
6
8
10

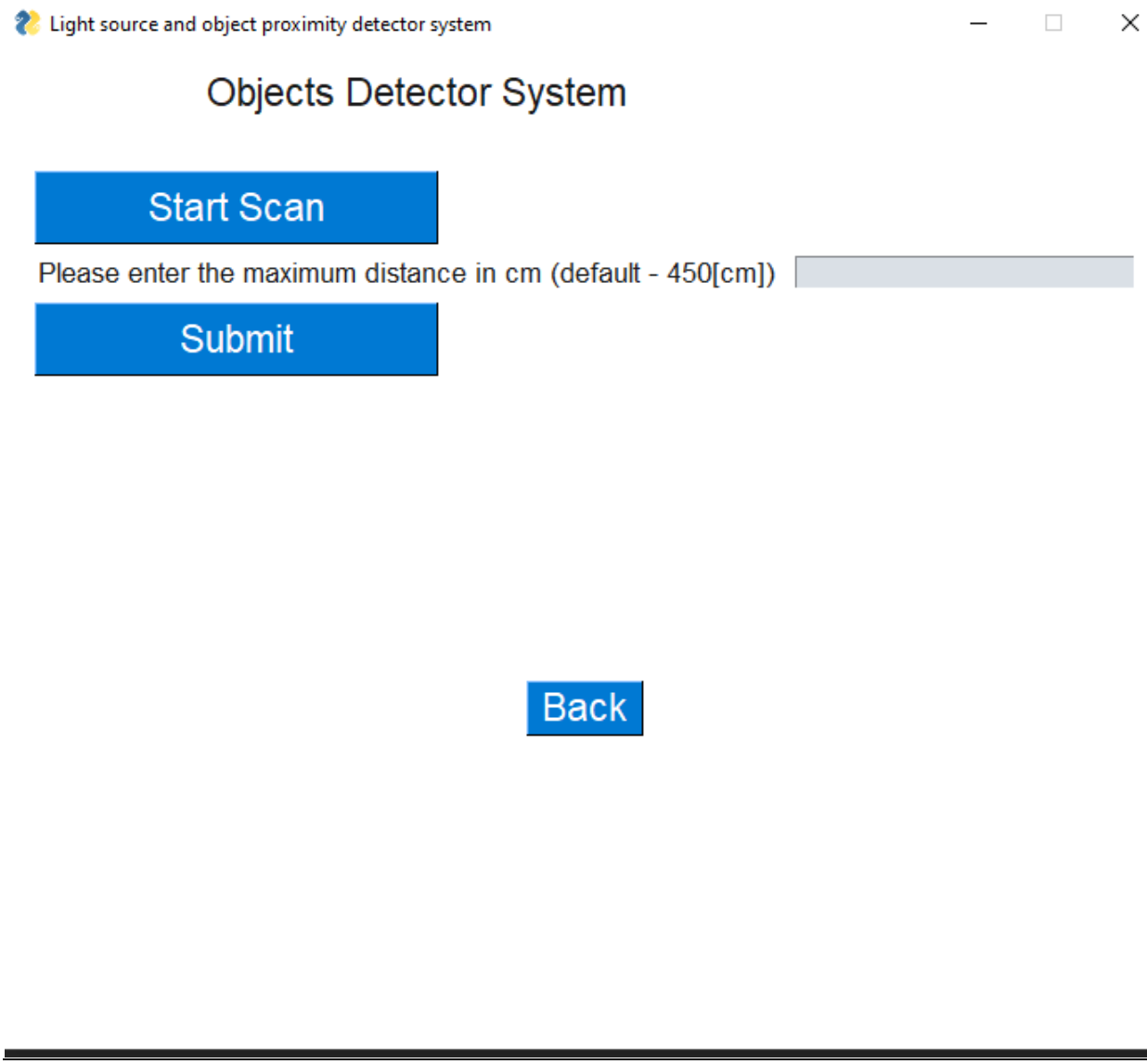


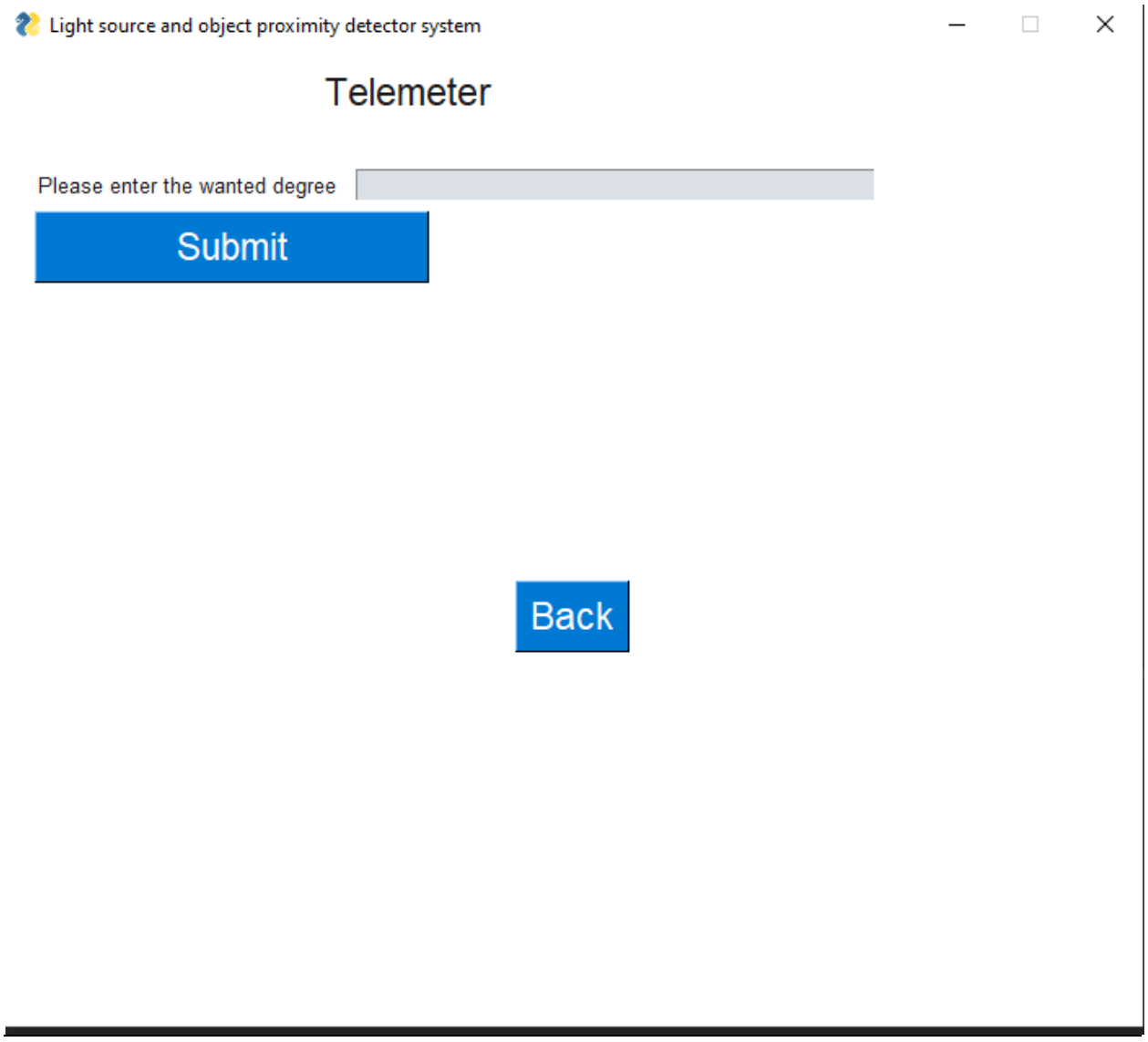

Waveforms can be exported into Excel/Wav/Osc
Spectrum Analyzer(FFT)
AC and DC Coupling
Cursor Measurements
Software support：Windows XP/7/8

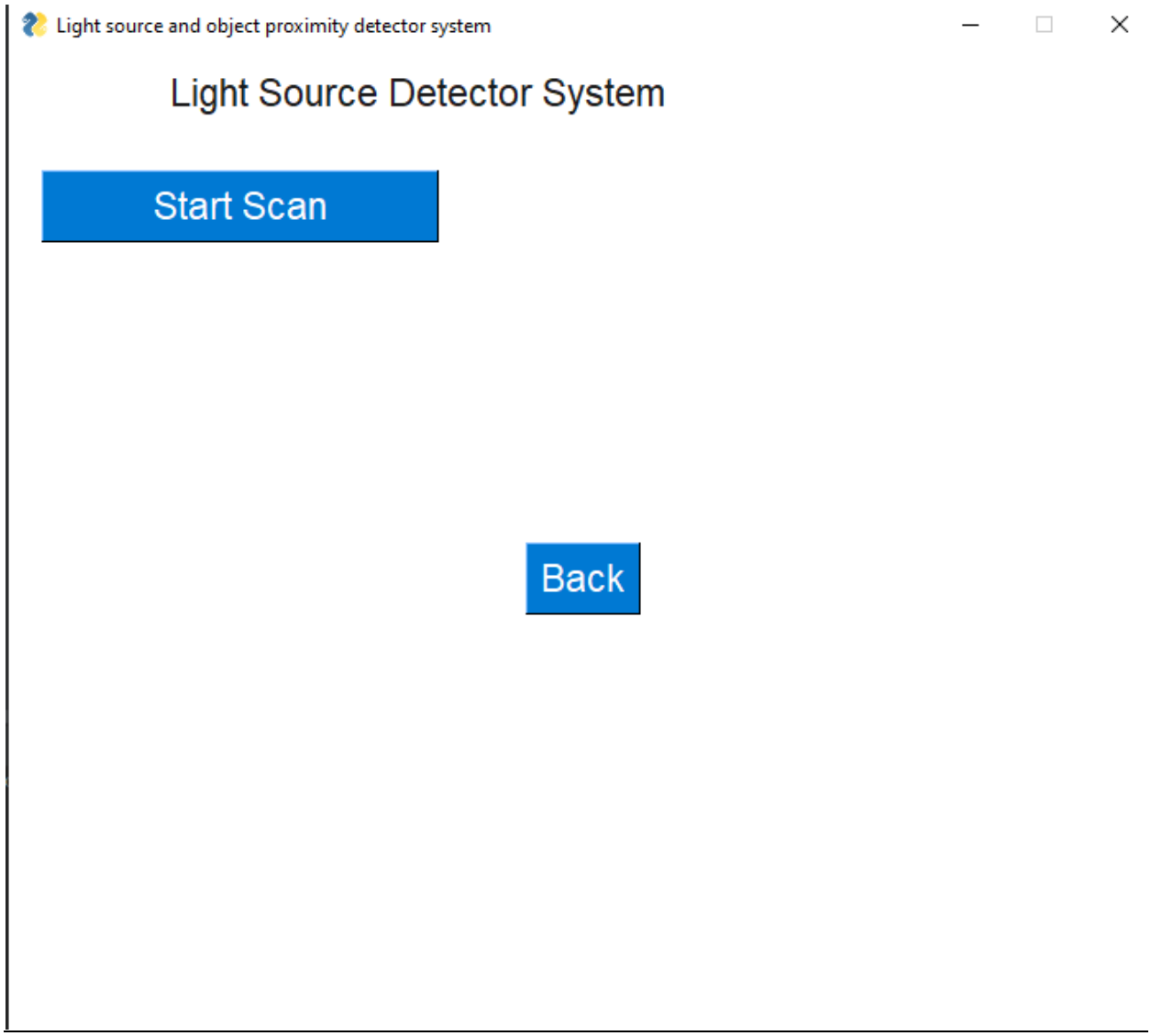


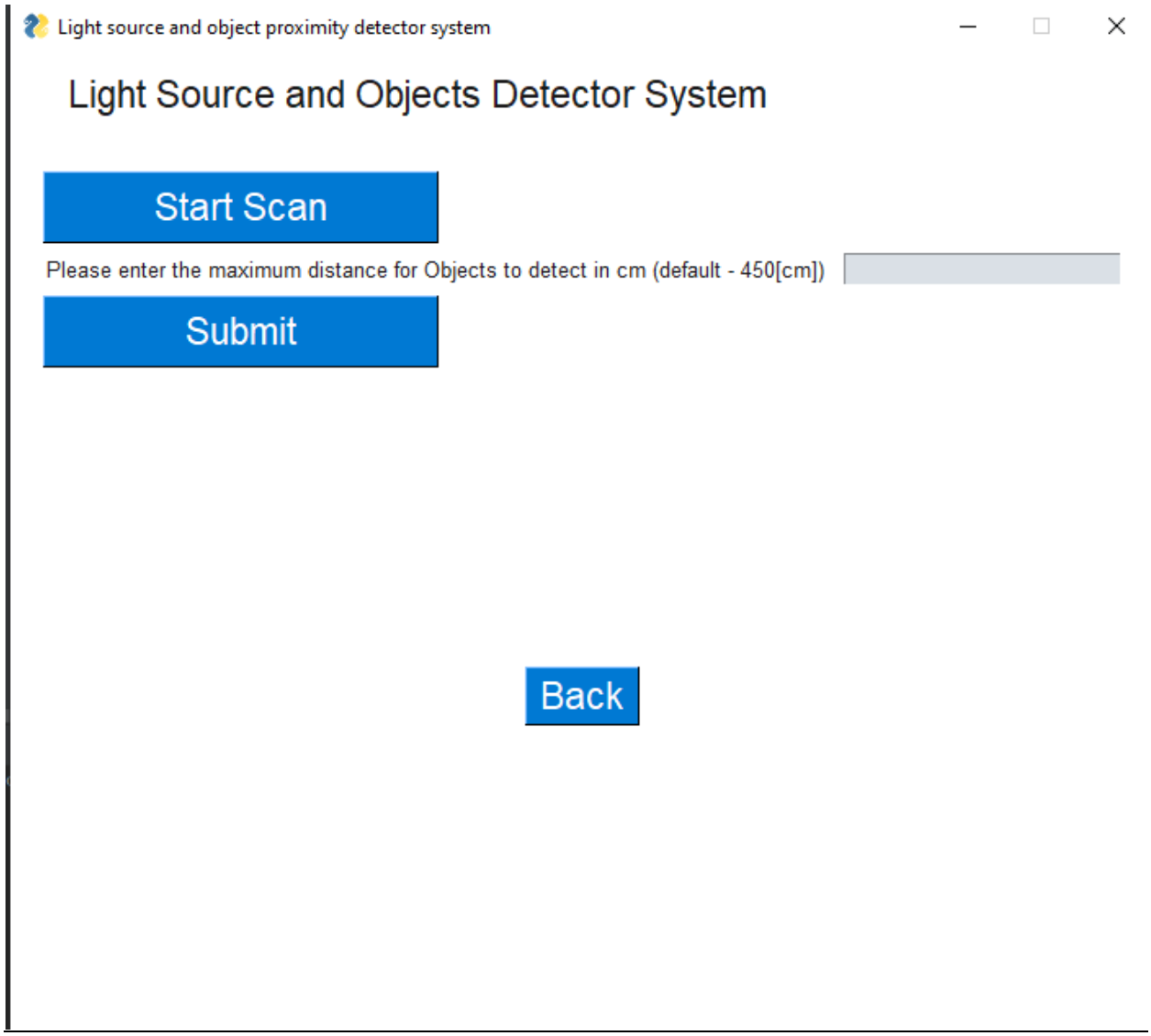
Application:

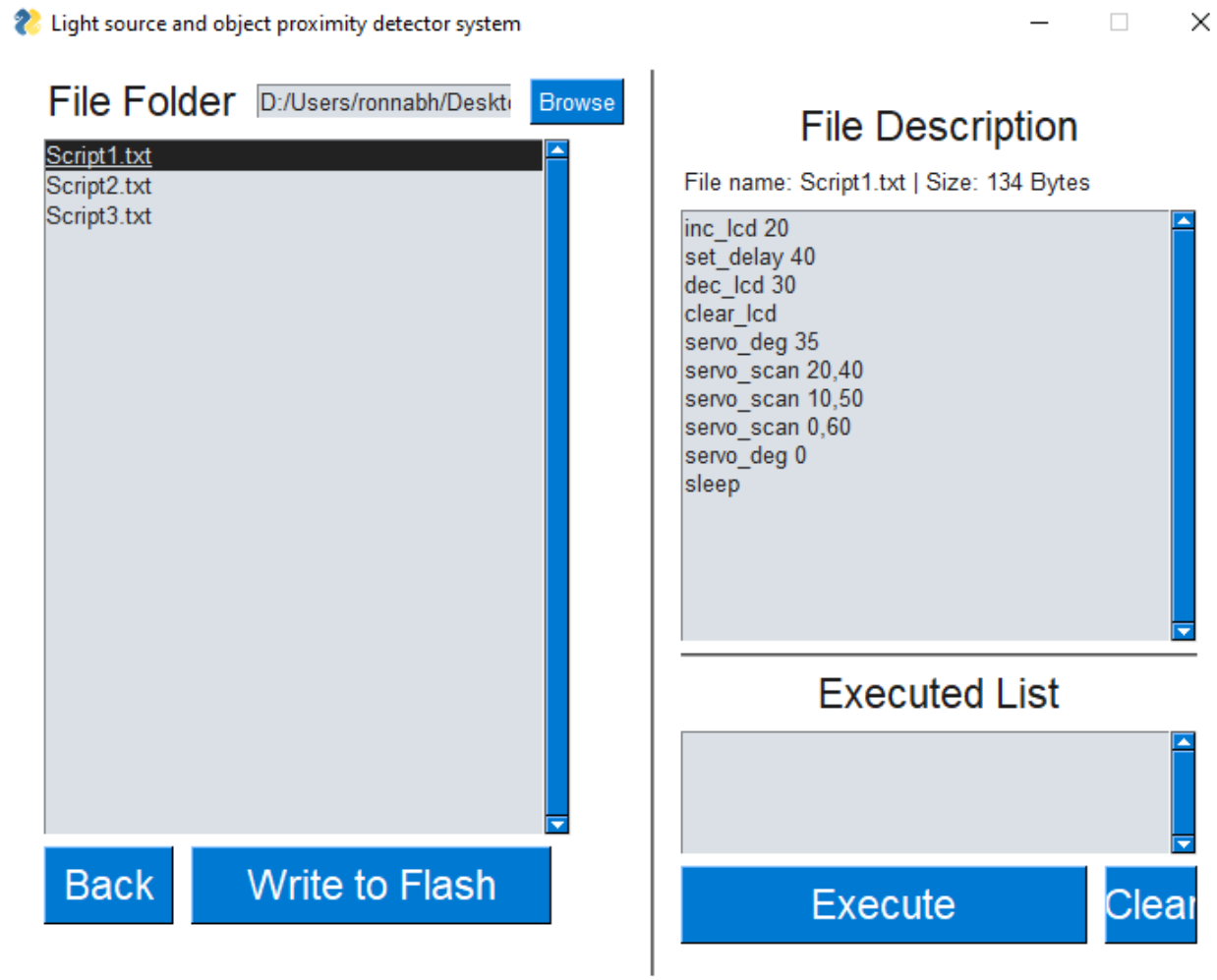

Light source and object proximity detector system
DCS Final Project - Ron and Matan
Objects Detector System
Telemeter
Light Source Detector System
Light Source and Objects Detector System
Script Mode
LDR Calibration


Light source and object proximity detector system
Objects Detector System
Start Scan
Please enter the maximum distance in cm (default - 450[cm])
Submit
Back


Light source and object proximity detector system
Telemeter
Please enter the wanted degree
Submit
Back


Light source and object proximity detector system
Light Source Detector System
Start Scan
Back


Light source and object proximity detector system
Light Source and Objects Detector System
Start Scan
Please enter the maximum distance for Objects to detect in cm (default - 450[cm])
Submit
Back


Light source and object proximity detector system
File Folder
D:/Users/ronnabh/Deskt
Browse
Script1.txt
Script2.txt
Script3.txt
Back
Write to Flash
File Description
File name: Script1.txt | Size: 134 Bytes
inc_lcd 20
set_delay 40
dec_lcd 30
clear_lcd
servo_deg 35
servo_scan 20,40
servo_scan 10,50
servo_scan 0,60
servo_deg 0
sleep
Executed List
Execute
Clear

OPC (first Byte)	Instruction	Operand (next Bytes)	Explanation
0x01	inc_lcd	x	Count up from zero to **x** with delay d onto LCD
0x02	dec_lcd	x	Count down from **x** to zero with delay d onto LCD
0x03	rra_lcd	x	Rotate **right** onto LCD from pixel index 0 to pixel index 31 a single char **x** (ASCII value) with delay d
0x04	set_delay	d	Set the delay **d** value (***units of 10ms***)
0x05	clear_lcd		Clear LCD
0x06	servo_deg	p	Point the Ultrasonic sensor to degree p and show the degree and distance (dynamically) onto PC screen
0x07	servo_scan	l,r	Scan area between left l angle to right r angle (once) and show the degree and distance (dynamically) onto PC screen
0x08	sleep		Set the MCU into sleep mode