# Final Project: Team A: IR Range Finder

*Colby Clark and Scott Wood*
Rowan University

December 22, 2018

## 1 Design Overview

The IR Range Finder uses an Infrared sensor that generates a voltage at a given distance due to reflected IR waves. The voltage generated would then be sent to the ADC of a microcontroller and converted to distance. Once the distance was calculated, it was sent to another microcontroller over UART that programmed an LCD screen to display the distance.

### 1.1 Design Features

These are the design features:

- 10 bit ADC
- IR photodetector with signal processor
- UART communication
  9600 Baud Rate
- LPM0 (Low Power Mode 0)

### 1.2 Featured Applications

Possible applications:

- Quick Measurement
- Sensor for robotics
- Object Tracker
- Smart Door

## 1.3 Design Resources

The code and README for the project can be found at the following link: `https://github.com/RU09342-F18/intro-to-embedded-final-project-team-a`

## 1.4 Block Diagram

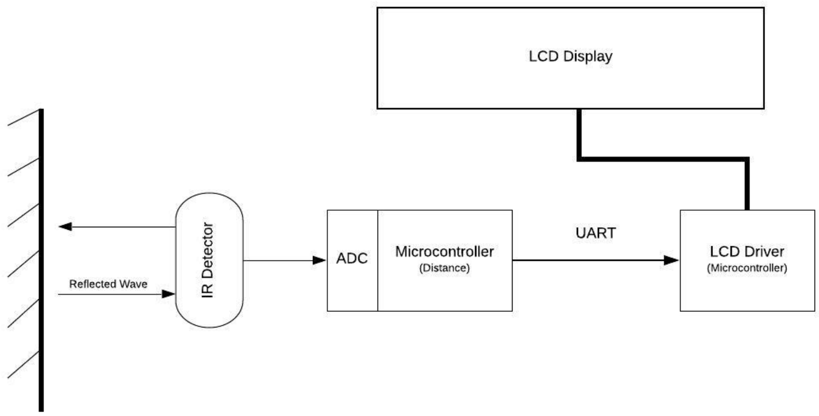


Figure 1: Block diagram for IR sensor

### 1.5 Board Image

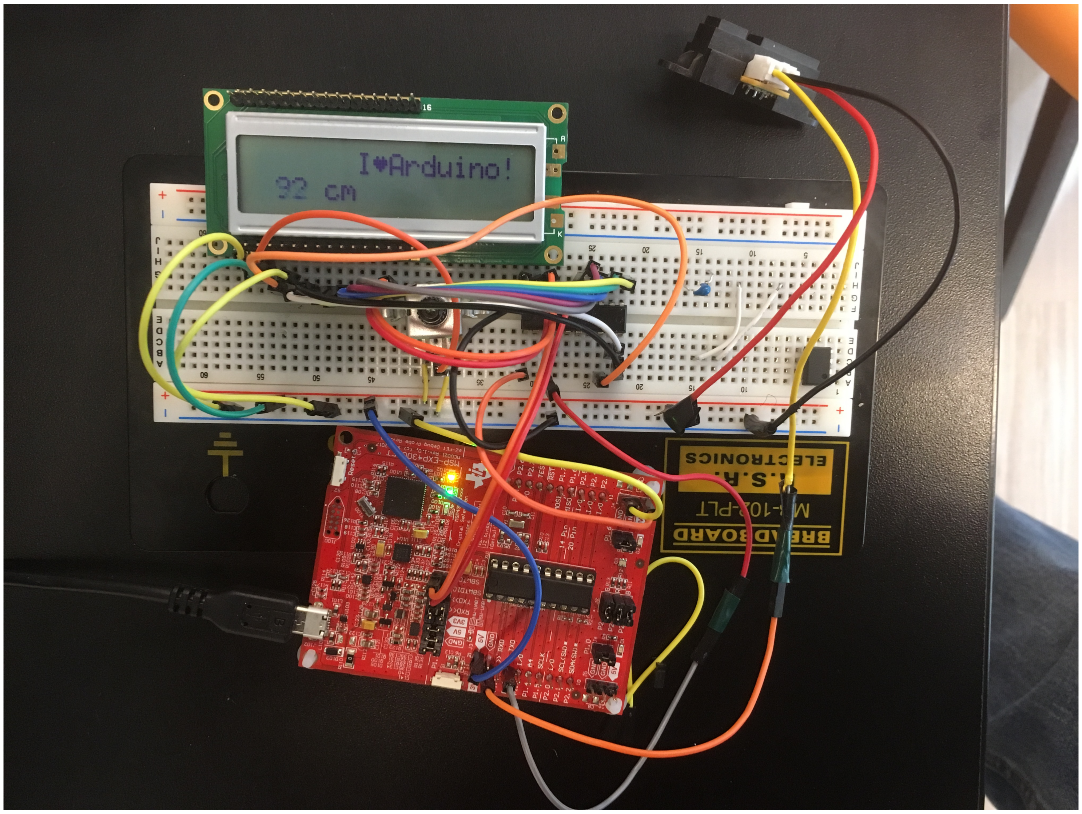


Figure 2: Photo of breadboard setup for IR rangefinder

## 2 Key System Specifications

| PARAMETER | SPECIFICATIONS | DETAILS |
|---|---|---|
| ADC Resolution | 10 BIT ADC | Smallest voltage the ADC can distinguish between is 3.2mV |
| ADC Reference Voltage | 3.3 Volts | Maximum input voltage to the ADC without losing data. |
| PIR Sensor | Sharp | 10 to 80 cm Measure Range |
| Surface of Reflection | Reflective Ratio of 90% | Results may vary with darker surfaces. |

# 3 System Description

Prior to the development of a rangefinder, people were limited to physical measurement tools. With the development of the R.F. one can now maintain precise and accurate measurements regarding the distance of an object at the press of a button. This information can also be easily stored digitally, making certain tasks in automation easier. Based on the functionality of the Rangefinder, the R.F device can be utilized to solve a variety of problems that regular motion sensors can not achieve. For example, a rangefinder can be used in Security Systems, Smart Doors, and object trackers. Despite the advantages, there are several external factors that come into play due to the technologies used. In the case of the IR sensor used for this project, electrical noise, and the reflective properties of material can play a shift in the distance calculated. Therefore, the conditions in which these sensors are used in have to be carefully considered.

## 3.1 Detailed Block Diagram

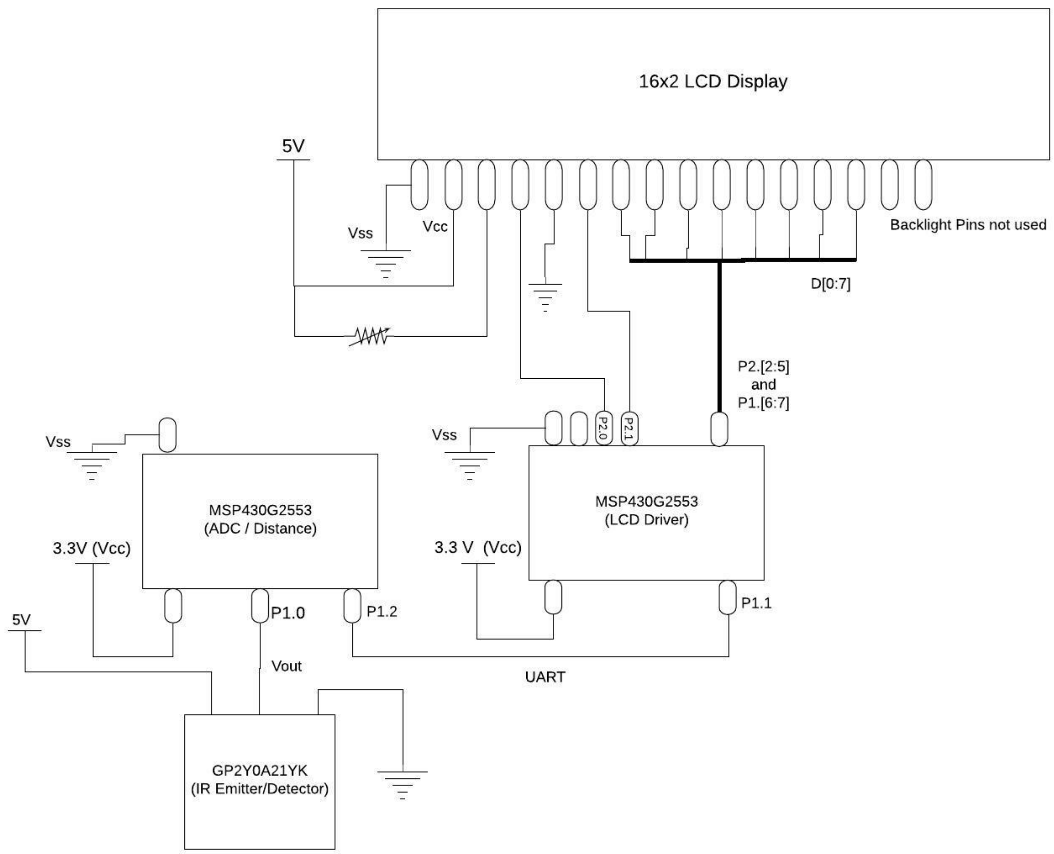


Figure 3: Detailed block diagram of rangefinder system

## 3.2 Highlighted Devices

- Infrared Sensor: Emits IR waves and uses a photodetector to pick up the intensity of the reflected waves at a given distance.
- MSP430G2553: Microcontroller. Two are used. One acts as an LCD Driver and the other takes the analog voltage generated by the IR sensor, converts it to a digital signal, and calculates distance. The two microcontrollers communicate with each other over UART.
- LCD Display: Used to show the current distance an object is from the IR sensor. Is a 16x2 digit design.

### 3.3 MSP430G2553

The MSP430G2553 microcontroller was chosen for this project because it had just enough GPIO pins to support the rangefinder, and so the system could easily be tested on a breadboard. Two of these microcontrollers were chosen to perform a specific task. One was used as an LCD display driver and the other converted the analog voltage received by the sensor to digital, and calculated distance. One LCD driver was programmed in Energia to save time, and the other "G2" was programmed using code composer.The two microcontrollers communicated with each other using a UART connection.

### 3.4 LCD Display

A 16x2 LCD display was selected for this project. To save time and reduce the overall programming complexity, the LCD driver for the display was programmed in Energia. The 16x2 LCD display works by repeatedly setting a cursor position, and then uploading a character to be displayed. The characters which are displayed are those stored sent to data pins of the display. Each row allowed for 8 bits to be displayed for the two provided line. The characters were sent in ASIC.

## 4 SYSTEM DESIGN THEORY

The basis of the rangefinder is the IR sensor. The GP2Y0A21YK0F sensor used, emits infrared waves that bounce off a surface and return to photodetector. When the photodetector recieves an IR wave it generates a voltage based on the intensity of the IR wave received. The intensity of the IR wave returned changes with multiple factors but mainly decreases the further it travels. To determine the distance over a given range, in our case from 10 to 100 cm, the voltage generated was recorded at discrete points. Below is the figure for the voltage characteristic as distance changes.

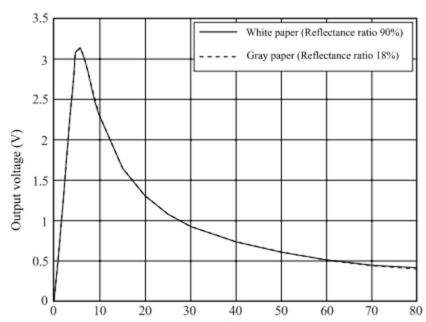


Figure 4: Voltage Characterization against distance

With the voltage curve, the distance at any point can be calculated within 100 cm. By sending the voltage value a given point to an ADC, the voltage value can be stored in a microcontroller. Once the microcontroller stores the voltage, it can calculate the distance and send it over UART to another microcontroller that acts as an LCD driver. The distance was sent in ASCII, so it was easier to program the position of the numbers received on to the LCD display. Once the distance is received by the LCD driver, it sets . The microcontrollers used for both task were the MSP430G2553, but the one acting as an LCD driver was programmed in Energia to save time.

### 4.1 Design Requirement 1: PIR Sensor With Filter

To measure the distance of an object in a set range, a PIR sensor was utilized and implemented. The PIR sensor selected for this project was the Sharp GP2Y0A21YK model. The selection of this PIR sensor enabled the effective range of measurement and distance calculation to be 10 cm to 100 cm. This PIR sensor is also equipped with an internal signal processing circuit, which was necessary to reduce the noise received when testing the sensor.

### 4.2 Design Requirement 2: Offboard MSP430G2553

To be able to communicate to the LCD display and then display the calculated distance on said LCD display, a MSP430G2553 IC had to be utilized. The MSP430G2553LP, which was programmed to calculate the distance was designed to communicate this calculated distance over UART to the MSP430G2553 IC, so that it could be display on the LCD display. Without the ability to program the MSP430G2553 IC on a breadboard this design goal and requirement would not have been achieved. The code necessary to do this was written in Energia to reduce the complexity in programming the MSP430G2553 offboard to drive a 16x2 LCD display.

## 5 Getting Started/How to use the device

The device and the overall system's functionality can be described in such a that anyone can use it and appreciate a device which calculates quick, precise, and accurate distance measurements. For example, as long as one possesses the required devices listed in the Highlighted Devices section, the source code listed in the Design Resources section, Energia imported in CCS, and the correct circuit configuration listed as Figure 3 the device should function with no issue upon compilation of the code. More detail for information regarding the specific steps for functionality of the device can be found within the subsections below.

### 5.1 Connecting the Device

The device only requires a simple connection to a PC with a micro USB cable provided with the MSP430 to program the MSP430G2553LP to drive the PIR sensor and communicate with the offboard MSP430G2553, which drives the LCD display. Once one completes this connection and verifies all the connections have made listed in Figure

3 and runs the program in CCS to program the MSP430G2553LP to drive the PIR sensor and runs the program in CCS written in Energiaa to program the offboard G2553, the device should be successfully connected and functional.

### 5.2 Circuit Setup

The circuit must be setup such that the MSP430G2553LP is connected to the PIR sensor, specifically the port, P1.0 must be connected to the yellow wire of the PIR sensor, which represents the signal output of the sensor. Power must be supplied to both the sensor and MSP430G2553LP such that 5V and 3.3V DC are supplied respectively. Furthermore, the MSP430G2553LP must be connected to the offboard MSP430G2553 such that UART communication is achieved. This requires that a connection between the MSP430G2553LP and the offboard MSP430G2553 be made from P1.2 to P1.1 respectively. The LCD display, which is driven by the offboard MSP430G2553 must be be provided 5V DC to Pin 2 of the display, and be connected with a 10 k *Omega* potentiometer to Pin 3, so contrast of the display can be controlled by the user. The following ports from the offboard MSP430G2553 P2[2:5] and P1[6:7] must be connected to pins D[0:7] of the LCD display for successful display of the data received.

## 6 Getting Started Software/Firmware

The program(s) required for the MSP430G2553 to calculate the distance values measured from the PIR sensor and the offboard G2553 for the LCD display were is written in C within an Integrated Development Environment (IDE) called Code Composer Studio (CCS). One would have to have CCS installed and the correct MSP430G2553 Family Device library selected to run and build the program on the MSP430G2553. The code for the LCD display was written in Arduino is was complied with CCS after Energia was properly imported. One would have to import Energia in CCS prior to compiling and running the LCD display code.

In order to successfully transmit data with the use of RealTerm, the proper UART settings had to be calibrated within RealTerm, which required a 9600 baud rate to be selected and set from the drop down menu labeled as Baud. This Baud Rate defined the rate at which the binary packets were transferred in the UART communication channel. Furthermore, the user has to ensure that the correct COM port is selected for proper communication to be achieved from their device to another or to their own device, in order to do this device manager must be pulled up and the USB PORT number, which is connected to the UART cable and visible in device manager must be entered and selected from the drop down menu labeled as Port.

### 6.1 Hierarchy Chart

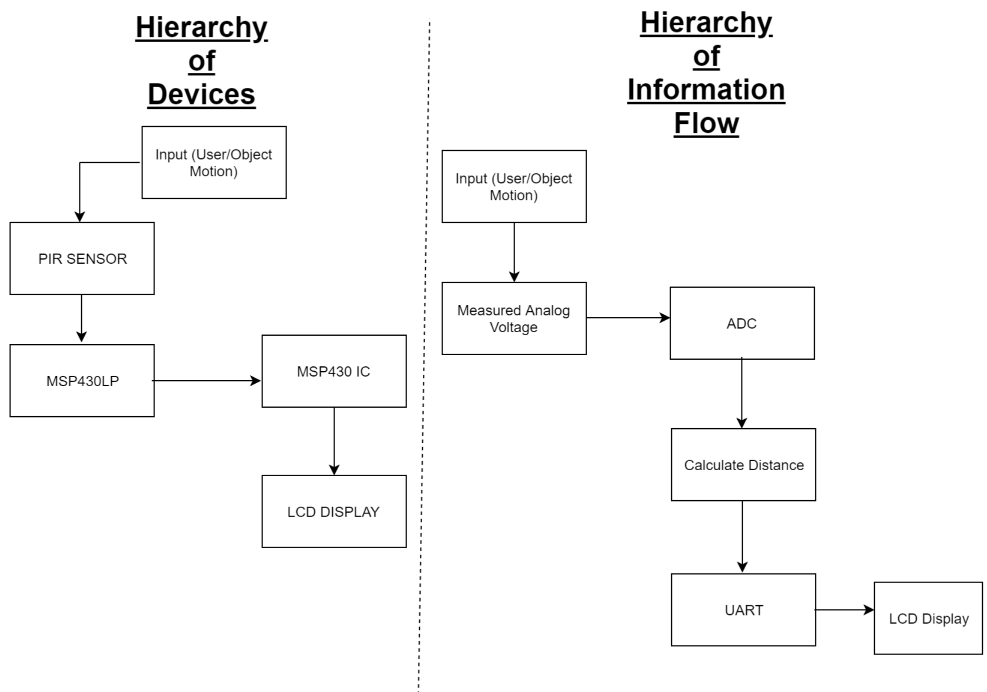


Figure 5: Hierarchy Chart of System

### 6.2 Communicating with the Device

Information can be sent to and received from the MSP430G2553LP using a micro USB cable and a terminal such as RealTerm. The calculated distance values will appear on RealTerm if one follows the proper setup. One can also send ASCII values to the LCD display with use of RealTerm with the proper setup. To achieve proper setup, prior to successfully receiving any calculated distance on RealTerm and after connecting the MSP430G2553LP with a micro USB cable to a PC, the user must open the PC's device manager and find the COM Port number of the connected board. This is found under *Ports (COM & LPT)* in the device manager. The desired COM Port number is that which follows the term *UART* in the list. After correctly setting this up one should be able to observe calculated distance values appear on RealTerm and verify that the values on RealTerm are identical to the ones on the LCD display.

## 7 Test Setup

To verify that the system is functional, set up the microcontroller, LCD display, and IR sensor by doing the following after programming. Connect the IR sensor to ground and a 5 volt power supply. After the sensor is powered, connect the output pin to Pin 1.0 of the first MSP430G2553. Pin 1.0 represents the ADC input. Then connect Pin 1.2, the UART Tx channel, to the Pin 1.1 (UART RX) of the second microcontroller that drives the LCD display. The LCD includes 16 pins, a register select pin, a read/write pin, and Enable pin, "8" data pins, Vss, Vcc, and two pins to power a backlight. The pins can be connected to the LCD display in any way, given that the pins are properly reconfigured in code. It is recommended that both microcontrollers be powered by 3.3 Volts. After configuring the system, confirm the accuracy of the results by comparing the distance from the rangefinder to a ruler. Make sure the rangefinder is pointing at a reflective surface for testing, by the reflective ratio of the object can alter the results of the rangefinder to a degree. The rangefinder is capable of measuring between 10 and 100cm accurately.

## 7.1 Bill of Materials

| Device | Cost | Quantity |
|---|---|---|
| Sharp GP2Y0A21YK PIR SENSOR | $13.95 | 1 |
| MSP430G2253 | $2.35 | 2 |

## 7.2 Radar Plot: Spiral Development

Listed below as Figure 6 is a highlight of each step required for the overall design and testing of the device such that it could be designed and developed in an efficient manner. For example, starting from the origin of the spiral to its outer edge, each line, which represents each step in the design and development process is assigned a numerical value, which highlights its importance in the project, and allowed for organization and an overall increase in efficiency in the project's development. Unfortunately, even with the efficiency that the Radar Plot provided and guaranteed some of the goals within the spiral were not achieved. This is not to be viewed as a failure because the main goal was achieved and a value of 70 was achieved within the Spiral, meaning, the system is able to effectively calculate the distance, communicate the distance to an offboard G2553 over UART, and display the distance to an LCD display. The values, which were not achieved within the Radar Plot include the range from 80-100. This is the range where if provided with more time, the system would have included a case, which enclose and allow the user to carry it with them and obtain quick distance measurements on the go. This would have also included and satisfied the goal of a PCB design. Within the range of 90-100, the system would have been designed to drive an alarm or an LED based off a certain calculated distance to act as a security system, and satisfy the requirement of a COB since to drive the alarm or LED would have required a Low Side Switch. Finally, at a value of 100 within the Spiral, the team would have implemented the ESP8266 Wifi module to display the calculated distance to a user over Wifi to their email or some other app of their interest.

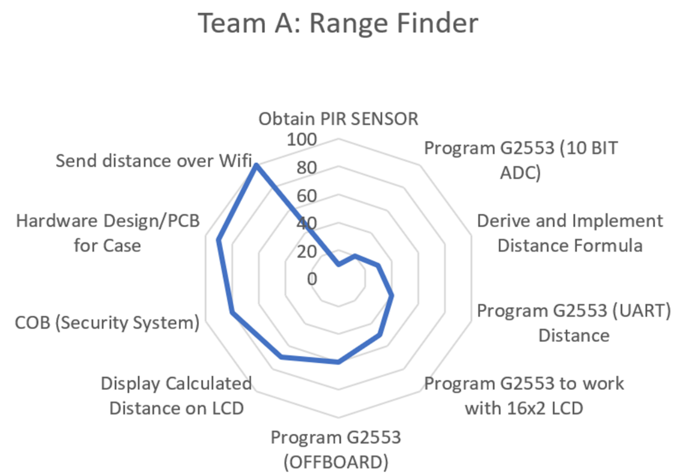


Figure 6: Radar Plot Highlights Progress and Objectives Achieved