# DETAILED DESIGN REPORT [CS 362]

# COURSE PROJECT: *PARKING LOT VACANCY*

## TEAM #16 MEMBERS

| **Member A** | | **Member B** |
|---|---|---|
| *Name*: David Hernandez | \|\| | *Name*: Juan Zambrano |
| *NetID*: dherna71 | \|\| | *NetID*: jzambr7 |
| *Email*: dherna71@uic.edu | \|\| | *Email*: jzambr7@uic.edu |

## ABSTRACT

This project will display available parking spaces in each lot. If one lot is taken the LCD screen will display some sort of indicator to go to the next lot. If there are free spaces said lot, then the LCD will display the parking lot position. For example, the LCD would print "`C is free`" so no one must look around for the parking space. This project would simulate real world situations were parking lots are busy and the possibility of time saved from the headache of looking for space. Parking in high density areas would be more efficient.

## DETAILED PROJECT DESCRIPTION

**Overall:** This project uses laser trip wires beaming from below to the embedded photoresistors above to determine which spaces available. The laser is directly aimed at a specific photoresistor for each space. When photoresistors receives light from the photoresistors it reads values of 800+. So, as soon as a car blocks the laser, the values from the photoresistor will fall dramatically and the code will determine it as space taken and only print spaces available via LCD screens – one for each level. The base wooden structure imitates a basic two-level parking garage, each level containing a respective floor and ceiling for necessary device installation. Arduinos microcomputers, most base wiring, and other miscellaneous devices are placed on the roof of the structure for most efficient maneuverability of wiring. LCD displays are hanged on the exterior roof of each level for reasonable visibility of viewing which spaces/levels are available.

**I/O:** In terms of input/output, as mentioned previously, photoresistors and laser trip wires are the most essential of the I/O devices of this project, namely input. Our project was designed to create two small miniature parking lots with each Arduino handling a parking lot of its own. We used wood to construct the two-level garage and toy cars to simulate this project. A laser trip wire setup consists of a laser aiming at a photoresistor. A laser trip wire is triggered when the photoresistor analog value decreases dramatically since there is no consistent light from the laser. When the laser trip wire is triggered, then we know that the space is taken. Each space taken the code handled any decrements and display only the open spaces on the LCD. The laser would be underneath the platform and the photoresistor would be on top of the platform. So, that way the laser can cut off when a toy car is station there. As far as wiring, this project will use a shield for each Arduino for any easy and cleaner setup. All the wiring was done on the shield extension, so every time we needed to disconnect either Arduino, we didn't have to painstakingly disassemble and rewire the necessary hardware, much to our convenience. Just plug and play.

**Communication:** The point of use of the communication between multiple Arduino is to handle each parking lot of its own. When the first level parking lot that Arduino is managing is full, the Arduino sent a signal to the other Arduino that is handling the second level via serial. The second lot will then display the parking space available. If one opens in the first lot, then the LCD in the second lot will turn off or displaying a different message and the LCD in the first lot will display the parking space available. In other words, the LCD in the second lot will only be active when the first parking lot is full for the sake of efficiency. This communication was achieved via serial.

## **ORIGINAL WORK**

Each lot will have three parking spaces with an LCD. Three laser trip wires each lot to detect any spaces available. When a car crosses the laser then space is taken. Once all three spaces are taken in level one lot then level two let will display any free parking spaces. As soon as one space is free on the first level parking lots, then level two lot will not display any free spaces available.

## **TIMELINE OF DEVELOPMENT**

- Week 1 – Learn how to communicate between two Arduino (Done)
- Week 2 – Code for one parking lot (Done)
- Week 3 – Test trip wire with one car (Done)
- Week 4 – Implement LCD (Done)
- Week 5 – Code for second parking lot (Done)
- Week 6 – Communicate both Arduinos via serial with new code (Done)
- Week 7 – Build two level platform parking lot and test for any bugs (Done)
- Week 8 – Implement a separate PCB that powers both Arduinos, lasers, LCD, & photoresistors (Done)
- Week 9 – Clean wiring (Done)

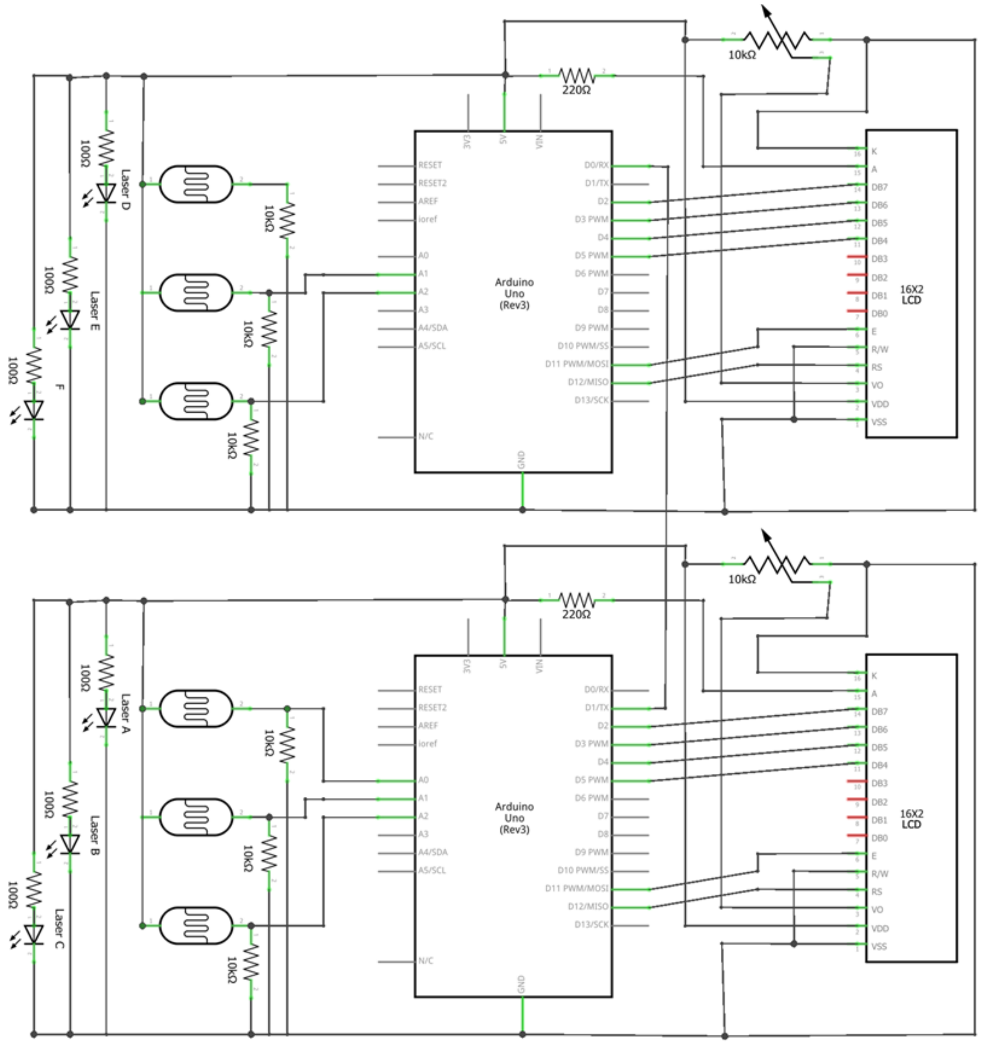
## LIST OF MATERIALS

- 6 Trip Lasers
- 6 Photoresistors
- Two Arduinos
- Two 16x2 LCDs
- 6 Toy Cars
- Wood for the platform
- Wires, Resistors, etc. (Electrical components)
- Shield for each Arduino [2x]

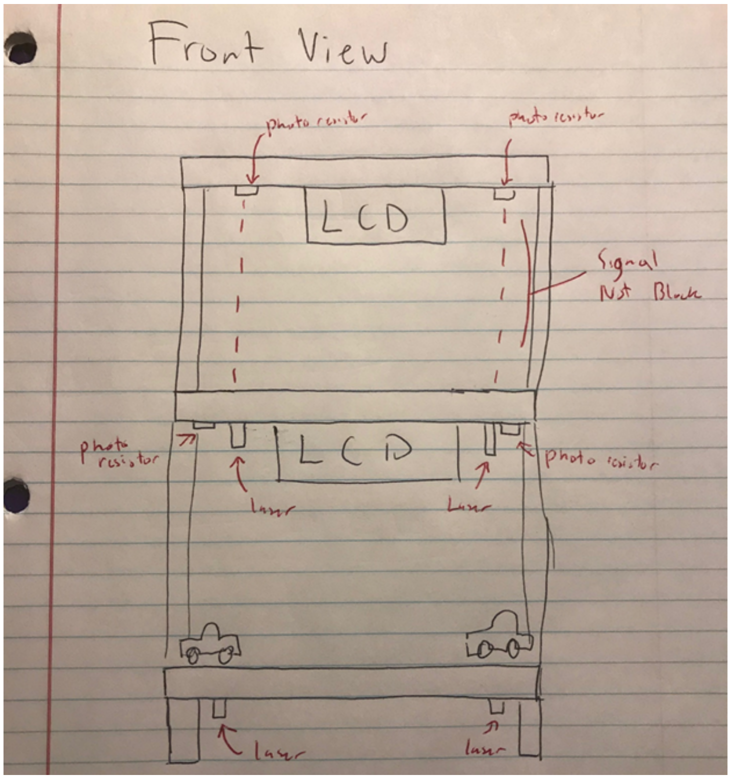
## LIST OF REFERENCES

- https://www.youtube.com/watch?v=BNfGoy5dqbg
  - Laser trip wire circuit and photoresistor tutorial
- https://www.instructables.com/id/How-to-use-a-photoresistor-or-photocell-Arduino-Tu/
  - Review for the photoresistor circuit
- http://playground.arduino.cc/Learning/PhotoResistor
  - Additional review for the photoresistor circuit
- https://www.arduino.cc/reference/en/language/functions/communication/serial/
  - Review of serial communication between Arduino devices
- https://www.arduino.cc/reference/en/language/functions/communication/serial/read/
  - Subsection of previous reference. Use of “serial read”
- https://www.arduino.cc/en/Tutorial/LiquidCrystalScroll
  - Review of 16x2 LCD circuit

## HIGH LEVEL DIAGRAM



## LOWER LEVEL DIAGRAMS



**<u>CODE</u>** **<u>w/ detailed commenting</u>** *(2 files)* https://github.com/jzambr7/ParkingVacancy_362

```
/* ParkingLot.h */

/* HEADER GOES HERE */
#ifndef ParkingLot_h
#define ParkingLot_h

struct space{
  int pin; // Pin for photoresistor on analog input
  int value; // value that holds the analog read in a photoresistor
  bool available; // If the resistor gets light then true, if no light then false
};

struct level{
  space A = {0,0,true}; // Declare a struct for space A
  space B = {1,0,true}; // Declare a struct for space B
  space C = {2,0,true}; // Declare a struct for space C
  bool dataLot; // Data for serial. true means spaces available, false means no spaces available.
} ;

level one; // Declare  a struct for level one parking lot
level two; // Declare  a struct for level two parking lot

#endif
```

---

```
/* ParkingLot.ino */
/* HEADER GOES HERE */

#include <LiquidCrystal.h> // Library for LCD
#include <ParkingLot.h> // Library for carrying info on Parking Lot
/* FIRST LEVEL LOT CODE */
// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);


void setup() {
  // put your setup code here, to run once:
  lcd.begin(16,2); // Setup LCD
  lcd.clear(); // Clear LCD
  Serial.begin(9600); // Begin Serial Communication
}

/* Scans any available spaces in the first floor */
void spaceScanner(bool &a, bool &b, bool &c, bool &x, bool &y, bool &z){
  one.A.value = analogRead(one.A.pin); // Read photo resistor for A
  one.B.value = analogRead(one.B.pin); // Read photo resistor for B
  one.C.value = analogRead(one.C.pin); // Read photo resistor for C
  two.A.value = analogRead(two.A.pin); // Read photo resistor for D
  two.B.value = analogRead(two.B.pin); // Read photo resistor for E
  two.C.value = analogRead(two.C.pin); // Read photo resistor for F



  /* if photoresistor recieves no light, then change Available to false */
  if(one.A.value <= 450){
    a = false; // Space in A is taken
  }
  else {
    a = true; // Space in A is free
  }

  if(one.B.value <= 450 ){
    b = false; // Space is taken
  }
  else{
    b = true; // Space in B is free
  }

  if(one.C.value <= 450 ){
    c = false; // Space in C is taken
  }
  else{
    c = true; // Space in C is free
  }
  if(two.A.value <= 450){
    x = false; // Space in D is taken
  }
  else{
    x = true; // Space in D is free
  }
  if(two.B.value <= 450){
    y = false; // Space in E is taken
  }
  else{
    y = true; // Space in E is free
  }
  if(two.C.value <= 450){
    z = false; // Space in F is taken
  }
  else{
    z = true; // Space in F is free
  }
}

/* Sends data to second floor depending if there are spaces available in first floor */
void lotFull(bool &a, bool &b){
  if(one.A.available == true || one.B.available == true || one.C.available == true){
    a = true; // Change datalot to true meaning spaces available in first floor
    Serial.write(a); // send data to second floor
  }
  else if(one.A.available == false && one.B.available == false && one.C.available == false){
    a = false; // Change datalot to false meaning no spaces available in first floor
    Serial.write(a); // send data to second floor
  }
  if(two.A.available == true || two.B.available == true || two.C.available == true){
    b = true; // Change datalot to true meaning spaces available in second floor
  }
  else if(two.A.available == false && two.B.available == false && two.C.available == false){
    b = false; // Change datalot to false meaning no spaces available in second floor
  }
}

/* Print spaces available on LCD */
void levelOneLCD(){
  lcd.clear(); // Clear screen
  int i = 0; // position for cursor


  /* Print on LCD depending if space is Available */
  if(one.dataLot == true){
    lcd.setCursor(0,0);
    lcd.print("Free Parking in"); // print

    /* Checks if A is available */
    if(one.A.available == true){
      lcd.setCursor(i,1);
      lcd.print("A");
      i = i + 2; // Move i two spaces to the right for cursor
    }


    /* Checks if B is available */
    if(one.B.available == true){
      lcd.setCursor(i,1);
      lcd.print("B");
      i = i + 2; // Move i two spaces to the right for cursor
    }
    /* Checks if C is available */
    if(one.C.available == true){
      lcd.setCursor(i,1);
      lcd.print("C");
      i = i + 2; // Move i two spaces to the right for cursor
    }

  }
  else{
    lcd.setCursor(0,0);
    lcd.print("GO TO NEXT LOT");
  }
}

void levelTwoLCD(){
  lcd.clear(); // Clear screen
  int i = 0; // position for cursor

    //Default message prior to receiving serial bit from 1st Arduino
    // Will display this message if LVL 1 has available spaces
    if (Serial.read() == true){
      lcd.setCursor(0,0);
      lcd.print("WAITING FOR");
      lcd.setCursor(0,1);
      lcd.print("LVL 1 TO FILL");
    }

    // If all spaces are occupied on LVL 1, then display spaces available on level 2
    /* Print on LCD depending if space is Available */
    else{
      /* If lot is not full, print available spaces */
      if (two.dataLot == true){
        lcd.setCursor(0,0);
        lcd.print("Free Parking in"); // print

        /* Checks if D is available */
        if(two.A.available == true){
          lcd.setCursor(i,1);
          lcd.print("D");
          i = i + 2; // Move i two spaces to the right for cursor
        }

        /* Checks if E is available */
        if(two.B.available == true){
          lcd.setCursor(i,1);
          lcd.print("E");
          i = i + 2; // Move i two spaces to the right for cursor
        }

        /* Checks if F is available */
        if(two.C.available == true){
          lcd.setCursor(i,1);
          lcd.print("F");
          i = i + 2; // Move i two spaces to the right for cursor
        }
      }
      /* If Lot is full, print "FULL" */
      else{
        lcd.setCursor(0,0);
        lcd.print("FULL");
      }
    }
}

void loop() {
  // put your main code here, to run repeatedly:
  spaceScanner(one.A.available, one.B.available, one.C.available, two.A.available,
two.B.available, two.C.available); // Scans any spaces available
  lotFull(one.dataLot, two.dataLot); // Send data to second floor
  if (Serial.available() > 0){ // Checks for any data received from serial
    levelTwoLCD(); // Print data in Level Two LCD
  }
  else{
    levelOneLCD(); // Print data in Level One LCD
  }
  delay(150);
}
```

**PICTURES OF PROJECT**