

# Parking System Strategy

Steps in developing the parking system:

**PAHSE 1:** Define the scope of the prototype

**PHASE 2:** Research

**PHASE 3:** Design circuit

**PHASE 4:** Acquire materials needed

**PHASE 5:** Prototype hardware

**PHASE 6:** Test hardware with software

**PHASE 7:** Integrate the system into our application

## PHASE 1: Define the scope of the prototype

The system will consist in 2 IR sensors which will simulate the possible scenarios of the parking spots availability. The system will be capable of tracking the status of the parking spots; if there are available or not. Their status will be sent to our application in order for it to react in the according way.

## PHASE 2: Research

In order to develop the application we needed to be familiar with the environments in which we'll work:

- the Arduino IDE along with the hardware
- the Java integration

**First**, we looked into the best alternative for input in our prototype. After investigating this matter, we concluded that infrared sensors are the best option.

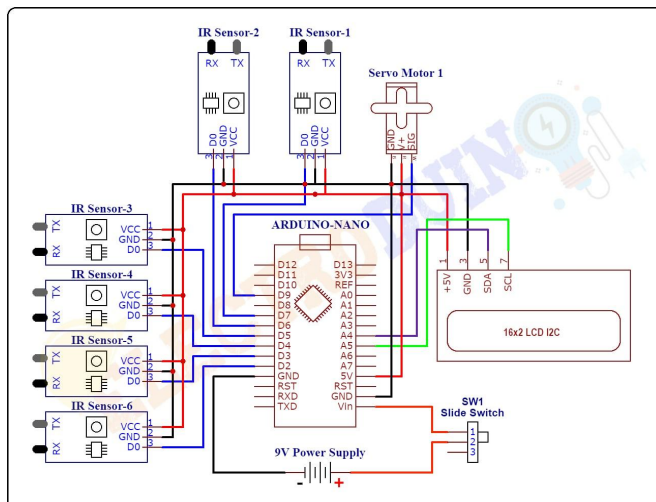
PROS	CONS
<ul style="list-style-type: none"><li>• easy to integrate</li><li>• simulates a real-life system</li><li>• the sensor detects motion in the daytime and nighttime reliably</li><li>• the sensor does not require any contact with the object to be sensed</li><li>• Infrared devices can measure the distance to soft objects which may not be easily detected by ultrasound</li><li>• no corrosion or oxidation can affect the</li></ul>	<ul style="list-style-type: none"><li>• objects (such as walls, doors) can alter infrared frequencies, as well as smoke, dust, fog, sunlight, etc. Therefore, it cannot pass through walls or doors</li><li>• it supports a shorter range and hence its performance degrades with longer distances</li></ul>

**Second**, we understood how the whole hardware system would look like: the sensors will be connected to the Arduino UNO using a breadboard and connection wires.

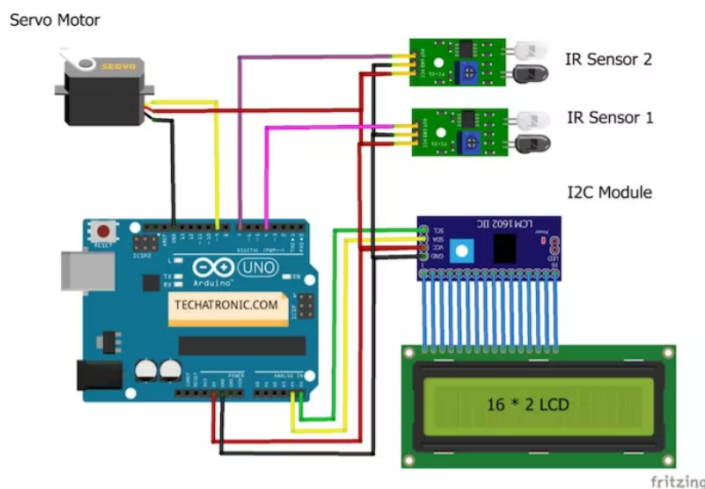
**Then**, we needed to find a way in which the Arduino could communicate with our Java code. We will be able to do this using a Serial Communication Library especially built for Java.

### PHASE 3: Design circuit

In creating our design, first we investigated the plans for similar parking systems. Because we'll integrate the system directly with our laptop in order to receive the signals, we won't need an extra power supply.



Also, adding an LCD Display might not be necessary but we could implement it for demonstration purposes.



#### PHASE 4: Acquire materials needed

- 2 IR sensors
- Arduino UNO
- Jumper Wires
- Breadboard
- LCD Display (maybe)

#### PHASE 5: Prototype hardware

- we are going to make connect the sensors and register their status
- we'll build an efficient way of transmitting the signals (only when their status change) and maybe add the LCD Display in order to observe these changes

#### PHASE 6: Test hardware with software

- ☐ does the state change for our parking spots?
- ☐ is the state updated in real time?
- ☐ how does our application handle the changes of state?

#### PHASE 7: Integrate the system into our application

The role of the sensors in our application is to notify the visitors if there is an available parking spot for them.

This means that the status of the parking lot will be sent to our application in order for it to take the actions that need to be taken depending on the situation: either send a confirmation that a spot is free or send the directions to the next parking lot available.

For testing this we will also need to integrate the API responsible for sending text messages.

## Resources

*Advantages and Disadvantages of Infrared sensor. (n.d.).*

<https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-Infrared-Sensor.html>

*Automatic car parking system project Using Arduino. (n.d.). Arduino Project Hub.*

<https://create.arduino.cc/projecthub/Techatronic/automatic-car-parking-system-project-using-arduino-ba2cb8>