Arduino Robot Lab 02

Smarter Movement with IR Sensors

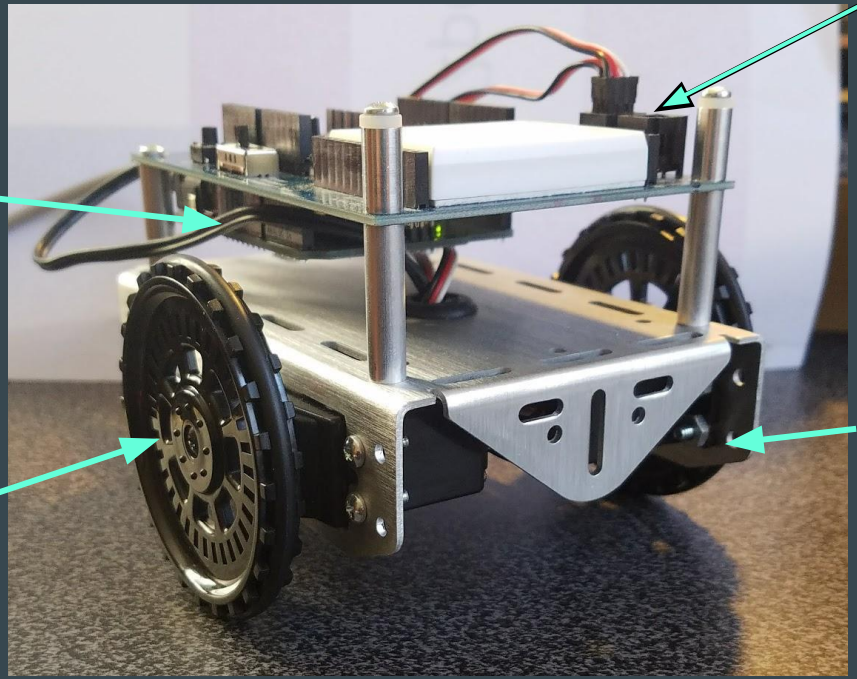
Zachary Collins / Isaac Daffron / Dom Farolino - Spring 2018

Recap

Let’s quickly go over what we did last time...

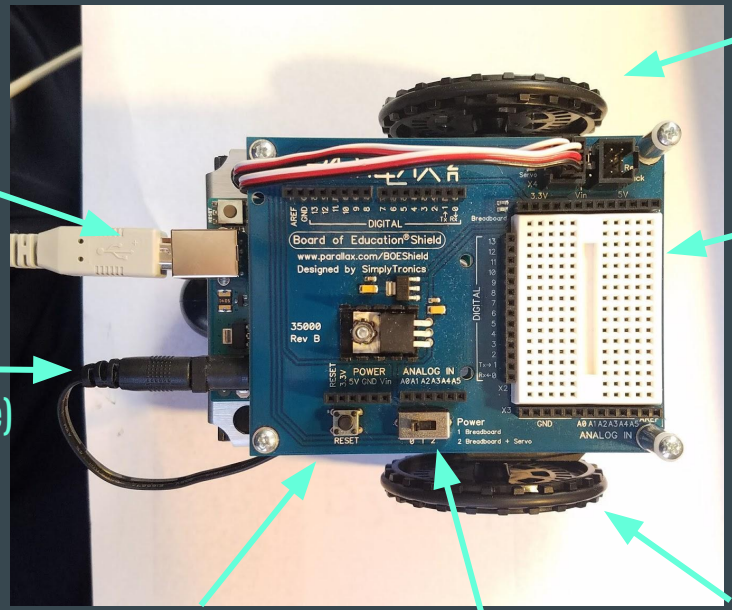
The Robot

Parallax Shield with Arduino



The Robot

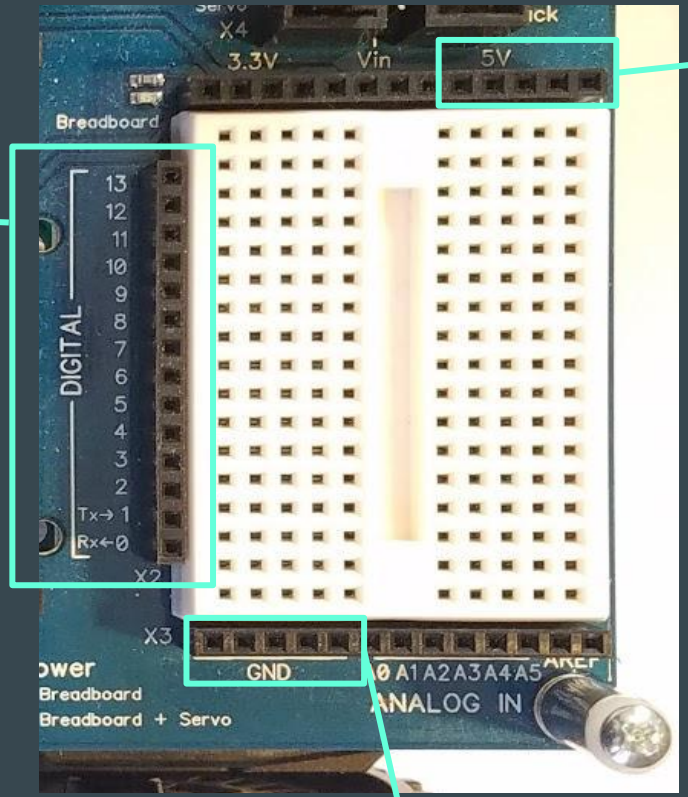
Hopefully you have a USB port on at least one of your computers!



Breadboard

This is where you will build your analog circuits.

Digital Inputs/Outputs
You can read/write 0 or +5V on these pins, corresponding to a digital zero or 1.



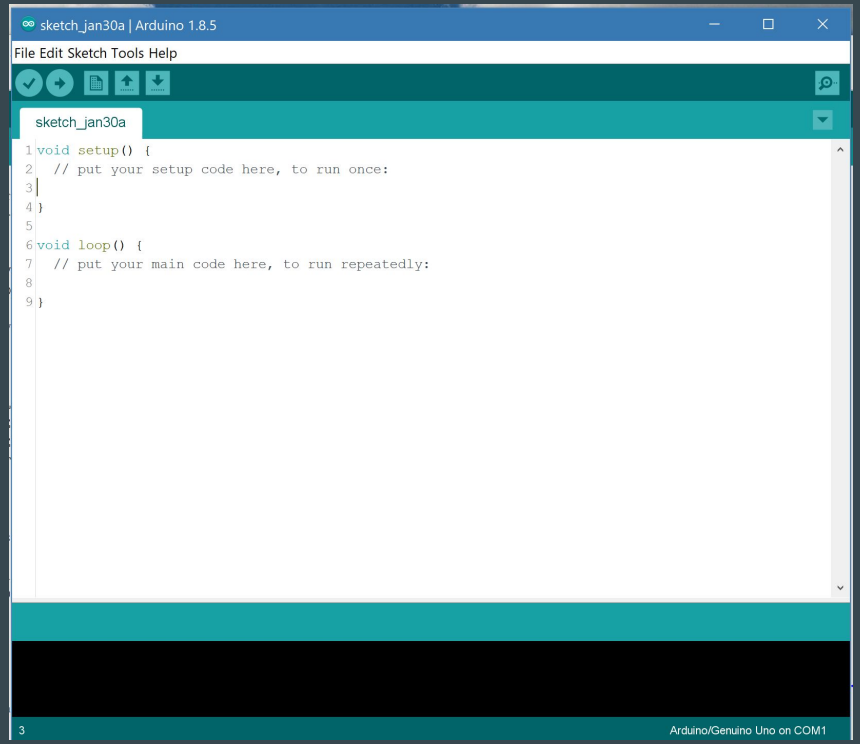
5V

Ground Reference

Arduino IDE

We will be programming using the Arduino's own IDE.

- Download at https://www.arduino.cc/en/Main/Software (aka just Google "Arduino IDE Download")
- Write code in "Sketches"
- These Sketches get uploaded to the Arduino board on the Robot.
- Every Sketch Has a setup() function And a loop() function. There is no main()! (that you can see).



Arduino IDE - setup()

- First function to run when the robot is turned on or reset.
- Initialize everything your robot needs to start running.
- Example:

sketch_jan30a | Arduino 1.8.5

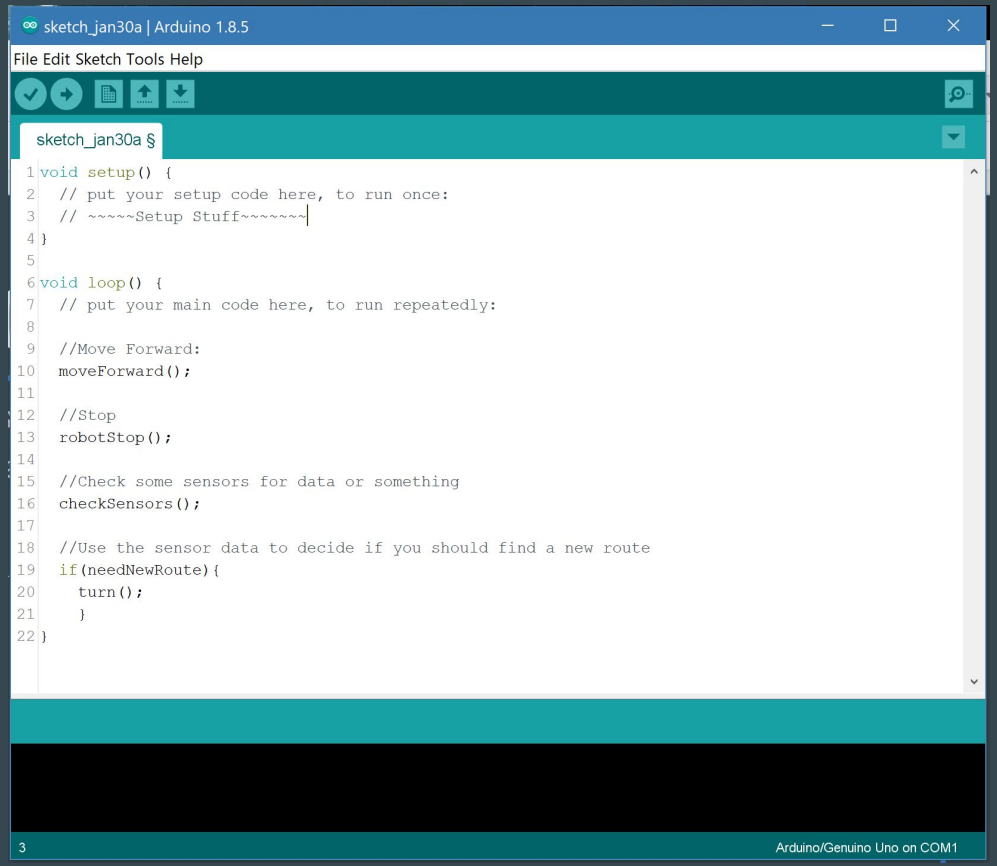
File Edit Sketch Tools Help

sketch_jan30a §

```
void setup() {
  // put your setup code here, to run once:

  //Open the Serial Port:
  Serial.begin(9600);

  //Attach the servos:
  attachServos();

  //Read ambient Infrared levels to use as a reference:
  readAmbientIR();

  //Maybe read some initial data from some other sensor:
  readSomeOtherSensor();
}
```

Arduino IDE - loop()

- loop() runs as soon as setup() is finished.
- It runs continuously, as long as the robot is on.
- Most of your code will be implemented in loop().
- Note that in setup() and loop() we should be calling lots of functions!

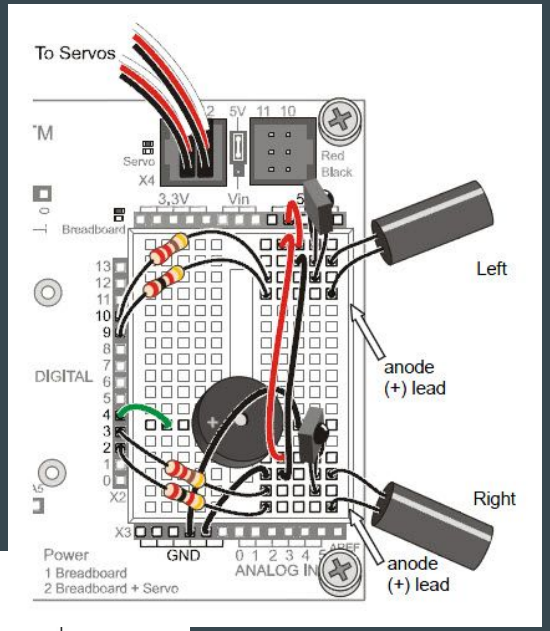


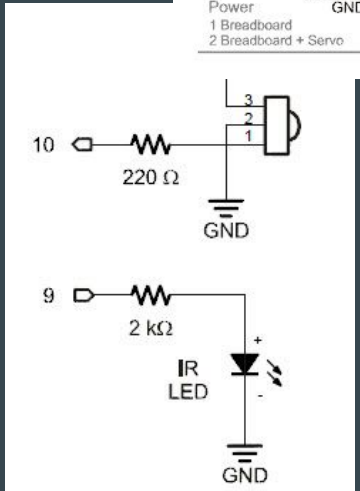
Infrared Sensors

- We can improve our robot by building circuits on its breadboard
  - Send output to pins
  - Read information from pins
  - Attach sensors to get information from the outside world

Infrared Sensor Circuit

- Build this circuit to the right (skip the buzzer)
- Circuit components
  - IR Receiver (2)
  - 220 Ohm Resistor (2) red red brown gold
  - 2k Ohm Resistor (2) red black red gold
  - IR LED (2)
  - Wires (At least 4)





Objectives

- Create a robot that steers itself
  - Read outside input
  - Make decisions about where to go
- Decision tree
  - Both sensors clear -> go forward
  - Right sensor blocked -> turn left
  - Left sensor blocked -> turn right
  - Both sensors blocked -> turn around
- Input filtering
  - Sometimes our sensors read imperfect data
  - E.g. 10 reads gives 1 1 1 0 1 1 1 0 1 1 when the sensor is blocked
  - Try to not let a few bad reads change your decision making

Pointers!

- Make all of your functions arguments pointers
- Syntax Review:
  - type *var_name; // Declares a pointer named *var_name*. Must point to an object of *type*.
  - &var_name // Outputs the address *var_name* points to.
  - *var_name; // Dereferences the value to which *var_name* points.
- Example
  - int my_int = 16; // Instantiates variable, *my_int*, as an int with value 16
  - int *my_pntr = &my_int; // Instatiates int pointer. Assigns it the address of *my_int*
  - cout << my_pntr; // Outputs address to which *my_pntr* points. Ex: 0x770fe5be40ec
  - cout << *my_pntr; // Outputs int value held at that address. In this case, 16.
- Use pointers as arguments of your functions
  - Dereference them within the function to use the value to which they point.

Skeleton Code and Grading

Code: https://github.com/zcollins0/robotics-practicum/tree/master/lab2

Basic functions are provided, but you'll have to write functions for turning and filtering input.

- Showing up: 10%
- Circuit built: 10%
- Robot movement (60% total):
  - Moving forward when you should: 30%
  - Turning based on sensor input: 30%
- Ignoring bad sensor input: 20%