2025-2026

Onboarding used for the 2025-2026 academic year

- Introduction
- Electrical
- Mechanical
- Software

Introduction

Hi there!

Welcome! We are super excited to have everyone here, and can't wait to introduce (or reintroduce) everyone to the amazing world of robotics. But first, we are going to start slow and get you comfortable with some of the basics.

What's Changed?

In the past, our onboarding has typically consisted of lecture style/interactive content where we would walk you through some of the processes we do in SCR. While this worked really well for some aspects of robotics, like the electrical side, it didn't do as well for others like software. This year, we are taking a different approach and throwing everyone straight into the deep end (with a life jacket of course).

The Deep End?

That's right, the deep end. You will be formed into small groups with the task of a building a small robot. While you will be building a simple line following robot that dumps a small bucket, the concepts you learn will stick with you throughout your time in SCR. The "Deep End" also happens to come with more inflatables in the form of written guides. Each week, you will be given a guide that focuses on a specific speciality which should hopefully give you a better understanding of our subteams.

- Week 1: [Electrical Basics & Soldering]
- Week 2: [CAD Basics & 3D Printing]
- Week 3: [Firmware Basics & Debugging]
- Week 4: Final Testing & Team Assignment

When is onboarding?

Onboarding spans from September 2nd to September 24th, taking place on each Tuesday/Wednsday from 6pm-8pm in the basement of REPF. You only need to attend one day each week, although you are more than welcome to attend both. If there are any changes we will let you know through our Discord, where all our announcements take place.

Will we have help?

Of course! Returning members, including our officer team, will be around to help and guide you throughout the process. We understand that learning something new can be overwhelming, especially in a new environment.

Do I need to do anything before hand?

Yes! Before we get started, you must have filled our your safety form here. If you join onboarding and participate in the activities, **we will assume** you have completed your safety form.

FAQ

- Q: Where is onboarding taking place?
  - We will be in the basement of the Rawl Engineering Practice Facility . Simply come in through the main entrance, take the stairs down, and walk through the doors.
- Q: I don't know anything about robotics, is that okay?
  - A: Of course it is! The point of "onboarding" is to introduce you to the wonderful world of robotics and all of its fun quirks from the ground up.
- Q: I cannot make it on one of the days, is that okay?
  - A: Yep! We designed onboarding to be two days a week where both days have the exact same content. We know schedules can be a bit tight, so hopefully having two days can allow everyone to attend!
- Q: Who all can go to onboarding?
  - A: Everyone! The only stipulation is that you must have completed your safety forms , you will need a photo of the front and back of your insurance card!

Electrical

Overview

Welcome to the Electrical Onboarding for the 2025-2026 school year!

Goals

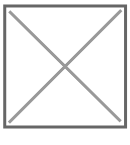
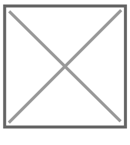
- Learning
  - What is a PCB (Printed Circuit Board)?
  - What is a Schematic?
  - How do you read a Schematic?
- Practical
  - Soldering components on a PCB
  - Debugging/Testing a PCB

Reading a Schematic

You can find the schematic for the PCB you will be working on here. The first three pages contain the actual schematic for the PCB, although you can mainly ignore the first page as it is purely for connections across sheets. A schematic is just a logical and visual representation of the PCB. The third (last) page of the pdf contains a top down view of the board with all of its layers.

The schematic is full of symbols like some of the ones seen below. alt text

You may also noticed that next to each symbol or component there is a designator, like SW1 shown in the image below. These correspond 1-to-1 on what you will see physically, as also shown below.

Understanding the Board

After taking a look at the schematic, you may have noticed there are quite a few components spread throughout the board. What do all of these do? How do they work?

> If you don't understand everything at the end of this session, that is okay! We don't expect you to learn everything about every part on this board, rather we just want to give you a general overview of what is going on.

Voltage Regulation

In order to actually power your robot, you will be using a battery holder that takes four AA batteries. With each battery at 1.5 volts connected in series (chained together), our max voltage we will have to deal with is around 6 volts. While this is great for some applications, our Pico is powered by 5 volts. To solve this, we can use something called an LDO which is found at designator **U1**. This will create a stable 5 volts from our unstable battery voltage. In our case, we only need this to power the Pico and then we use the 3.3 volt output (it has a LDO built in) on the pico to power everything else (explained later).

At designator **D1** you'll find a diode. This essentially ensures that power can only flow one direction, in the case of this diode it only flows towards the direction of the silver band (indicated in the picture below). The reason the board requires this is because when you have your computer plugged into the Pico (which provides power across the VBUS line) and the batteries connected, you don't want the battery power flowing back into your computer.

Raspberry PI Pico

The Raspberry PI Pico is our chosen microcontroller for this project. A microcontroller is more or less just a tiny computer that is used to interface with hardware components like motors, servos, sensors, etc. You will be given a presoldered pico for your robot.

Line Sensors and Servos

Both of these are fortunately pretty simple. The Line Sensors are powered via 3.3v from the Pico and have their data lines connected directly to the Pico as well. These are simply read in firmware by checking whether the signal is Low or High. The servos are powered straight from the battery and are driven directly off of the Pico.

Motor Driver

In order to control the motors in a safe and reliable way, we will make use of a Motor Driver. This small IC (Integrated Circuit) found at designator **IC1** takes the control signals that the PICO produces and converts them to the power needed to drive the motors. Other than some capacitors nearby per the datasheet, there isn't much else here!

LEDs

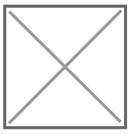
The LEDs are pretty basic, you connect power and ground and they light up. Most importantly, you will see that there is a resistor before each LED. The purpose of these is to limit the flow of current into the LEDs which helps prevent burnout or any issues. They are polarized, which the long leg of the LED indicating the positive side, which is also indicated on the board.

Capacitors

To keep it simple for now, all of the capacitors on the board are meant to supply charge at a moments notice. Additionally, they help to filter out high frequency noise. Basically, in general you want these by most IC's and outputs.

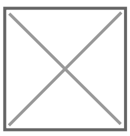
Soldering Basics

Before we get to assembling, lets walk through a few rules and tips about soldering. But first, what is soldering? Soldering is a method to permanently attach a component to something like a printed circuit board. Typically, as shown in the graphic below, you use solder (a metal alloy) and melt it to create that connection.



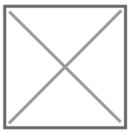
Step 1: Tinning

This is a step that helps protect the tip of the soldering iron and can help in general use of transferring solder to the component/board. Start by letting the soldering iron heat up, clean the tip with the provided sponge, and then press some of the solder on the tip and let it flow for a moment. It doesn't need a lot, just enough for the tip to turn a similar color to the solder.

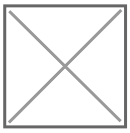


Step 2: Soldering

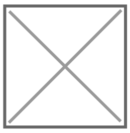
Well, there are only two steps here but this is a really big step. Lets start by mounting the component that you want to solder to the board. On the top side, place it through the board in the correct orientation (if you are confused here please ask a mentor) and bend the legs such that it won't fall out when you tip the board over.



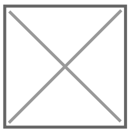
Next, lets heat up the pad of the area you want to solder. The pad is just the small metal circle (or square), known as a solder joint, that surrounds the hole.



Finally, lets apply some solder to that pad. Continue holding the soldering iron on that solder joint and touch your solder onto it (the solder joint) at the same time. Try not to touch the solder directly to the tip of the iron at this point, the solder joint should be hot enough to melt the solder.



You should end up with something that looks more or less like this! We will also try to wander around and verify your first solder to make sure you are on track!



Important Notes

- **Do not** touch yourself or any other person with the metal tip of the soldering iron at any time. These irons get extremely hot and will cause harm.
- Whenever you finish with a soldering iron, be sure to clean the tip!
    - This includes when you are passing it around between people. It can help keep your solder joints clean :)

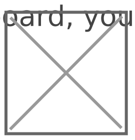
PCB Assembly

Now that you have learned some of the basics on how to solder and what all of the components do, its time to start assembling! For all of those in your team that wishes to solder, you should try breaking up the components so that each person gets to solder one of each component as practice! You may not be able to finish your entire PCB. Below is a full list of components in the approximate order they should be soldered:

1. D1 - Diode (silver ring towards LEDs)
2. C(4/5) - 10uF Capacitors
3. C(1/2/7/8/9/10/11) - 0.1Uf Capacitors
4. C(6/12) - 100uf Capacitors (long leg towards trace)
5. R(2/3/4/5) - 820 resistor (the tiny ones)
6. R1 - 10k resistor (the big one)
7. LED(1/2/3/4) - LEDs (any color)
8. J1 - Screw Terminal
9. U1 - LDO (will be given once all of the above are soldered and checked)

> **Don't be afraid to ask for help!**
> Soldering for the first time can be a scary process, but we are here to help you! If you and your team do not feel comfortable soldering your board, that is perfectly fine as well!
>
> Most importantly, if you get burned in any way **immediately** inform one of the officers so we can assist you as needed.

The Finished Product

After assembling your board, you  should have something like below (the space in the centre is where the pico will be).

Testing

Now it is time to test your PCB! If there is enough time for this, contact one of the mentors and we will walk through the testing steps. If there isn't enough time, we will manually test your PCB to ensure all of the individual components work. We will test your PCB regardless of whether we got to test it in the session just to double check :)v

Mechanical

Overview

Welcome to the Mechanical Onboarding for the 2025-2026 school year!

Goals

- Learning
  - How do we use CAD software in SCR?
  - What is a part file?
  - How can you import part files?
  - What is an assembly file?
- Practical
  - How to use Onshape
  - How to design parts for 3D printing
  - Export, slice, and print 3D models

Getting Started

Create an onshape account

Navigate to onshape and create an account. Select education, student, and college. Choose the student plan and search for the **University of Oklahoma**.

Joining The Onboarding Team

Send your onshape account email in the onboarding-mechanical discord channel.

Once you've been added to the team you should be able to see the shared parts folder for onboarding.

Creating Parts

This tutorial will go through the steps for creating a payload bucket.

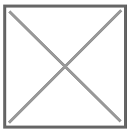
and create a document for your parts in your teams onboarding folder

Open the document and select **Sketch**

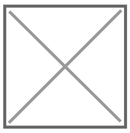
Choose the top plane to start your sketch. You can hide the front and right planes while editing this sketch like this

Create a **center rectangle** and dimension it at 1.85in on each side.

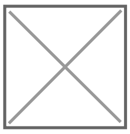
Select **extrude** and extrude at a depth of 1.85in.



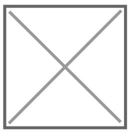
Select the 8 edges of the part and create the following fillets

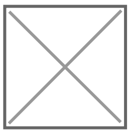


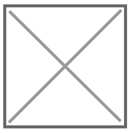
Create the following shell on the top face of the part.



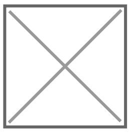
Create the following sketch and dimensions on the front face of the part



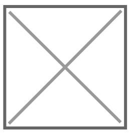




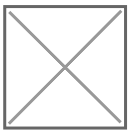
Select the circular pattern tool



Select the outer circle and move the origin of the pattern to the inner circle.



Change the count of the pattern to 4 and the angle of the pattern to 270.



Exit the sketch and select the **extrude** tool, then choose **Remove**, **Up to next** and select the back of the front face of the part.

The final part should look as follows:

Follow Along

Navigate to **SCR Onboarding 2025-2026 shareable** and copy the **3D Prints** workspace to your team's folder.

Edit the components based on the engineering drawings below.

Note: The provided variable table may be helpful for convenience.

Base

Enclosure

The team name is a **.025in** cut on the enclosure surface

Line Follower Mount

Software

Overview

Welcome to the Software Onboarding for the 2025-2026 school year!

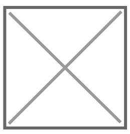
Goals

- Learning
  - How do we write software in SCR?
  - How does a microcontroller control hardware?
- practical
  - Writing Arduino/ C++ code
  - Managing software dependencies
  - Reading software documentation

Downloads

Arduino:

Go here and download the version of the arduino IDE for your operating system (Windows, OSX, linux).



Pico board support:

We'll be using the Raspberry Pi Pico 2 for onboarding. To develop on the pico, you need to add an additional board manager URL to the Arduino IDE.

Go here and scroll to the **installation** section. Follow the short directions there.

In the boards manager tab, search for these two libraries and **install** them

Compilation

Select the Pico 2:

Use the Arduino IDE to select the Pico 2 as the compilation target.

Connect the Pico 2 to your computer using a micro-usb cable. You should see this drop down turn **bold**.

Verifying Source Code:

Copy this skeleton code into an arduino sketch file.

```
#define LED1_PIN 13
#define LED2_PIN 14
#define LED3_PIN 15

#define IR1_PIN 0
#define IR2_PIN 1
#define IR3_PIN 2
#define IR4_PIN 3
```

```
#define IR5_PIN 4

#define MOTOR2_IN3 18
#define MOTOR2_IN4 19

#define MOTOR1_IN1 20
#define MOTOR1_IN2 21

#define SERVO_PIN 22

Servo payloadServo;

void driveForward() {
  // right motor
  digitalWrite(MOTOR1_IN1, HIGH);
  digitalWrite(MOTOR1_IN2, LOW);

  // left motor
  digitalWrite(MOTOR2_IN3, HIGH);
  digitalWrite(MOTOR2_IN4, LOW);
}

void dumpPayload(Servo servo) {
  servo.write(180);
}

void setup() {
  // IR Pin Setup
  pinMode(IR1_PIN, INPUT);
  pinMode(IR2_PIN, INPUT);
  pinMode(IR3_PIN, INPUT);
  pinMode(IR4_PIN, INPUT);
  pinMode(IR5_PIN, INPUT);

  // LED Setup
  pinMode(LED1_PIN, OUTPUT);
  pinMode(LED2_PIN, OUTPUT);
  pinMode(LED3_PIN, OUTPUT);
  pinMode(LED_BUILTIN, OUTPUT);
```

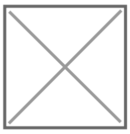
```
  // Motor setup
  pinMode(MOTOR1_IN1, OUTPUT);
  pinMode(MOTOR1_IN2, OUTPUT);

  pinMode(MOTOR2_IN3, OUTPUT);
  pinMode(MOTOR2_IN4, OUTPUT);

  // Servo setup
  pinMode(SERVO_PIN, 22);
  payloadServo.attach(SERVO_PIN);
}

void loop() {
  // put your main code here, to run repeatedly:
  driveForward();
  delay(500);
  dumpPayload();
  delay(500);
}
```

This file contains `#define`s for the Pico 2's inputs and outputs as well as a function to drive both motors forward.

Edit the `setup()` function and add the line

`digitalWrite(LED1_PIN, HIGH);`

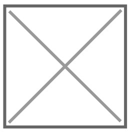
Click the check mark at the top left of the IDE to check for compilation errors. If your program compiles correctly you should see this output.

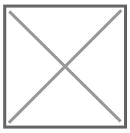


Flashing Pico 2:

Click the arrow at the top left of the IDE to flash the Pico 2. If flashing doesn't work, you may need to unplug the Pico 2 and hold the BOOTSEL button while reconnecting the micro-usb cable.

If your flash is successful, you should see this output, and LED1 should be turned on.

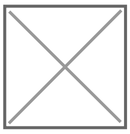




Useful Documentation

Motor Driver:

You shouldn't need more than this truth table, but if you'd like to read the full documentation you can do that here.



Arduino:

Arduino API, Servos, Search