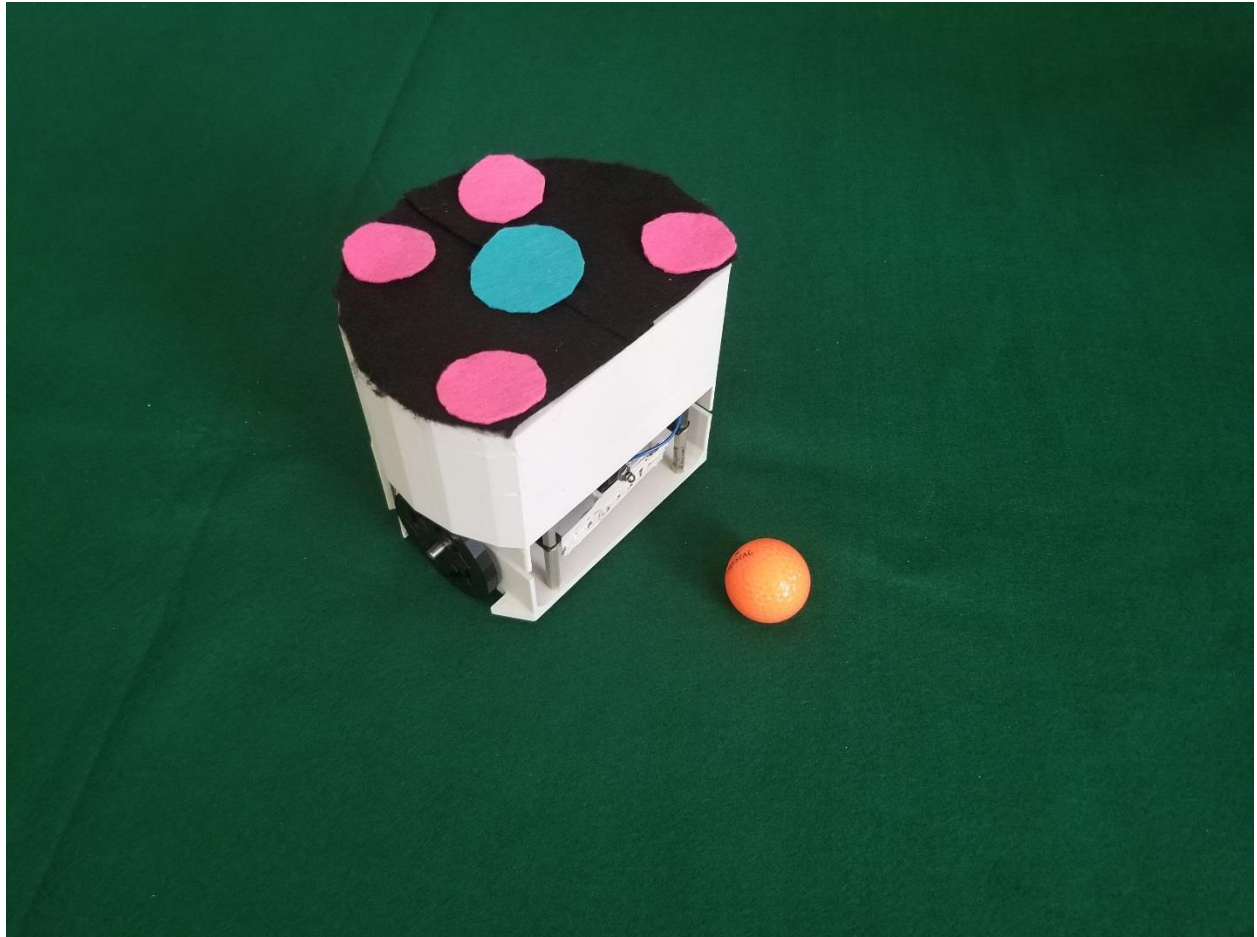


ROBOCUP AT UNB

User's Guide



Koceila Cherfouh Mike Trainor Eric Henderson Guovue Yan



Setup

Field of Play

The field of play is a 2x2 green felt surface. The color and material creates a low glare, high contrast background for the robots. The field must be placed on a flat surface for consistency and safety. Ensure that the felt surface is as smooth as possible, does not have any creases, and is free of debris.

The camera is mounted to a PVC frame mounted above the center of the field of play. Ensure that the frame and camera are both secure. If the camera cord is too loose and is visible within the camera frame, consider taping the cord onto the PVC frame.

PVC Frame Setup

A couple things you will receive inside along with the package to set up the frame for the Logitech camera. Those include two elbow joints, three 1.5-meter-long segments of $\frac{3}{4}$ " PVC pipe, 2 T-connectors and 4 shorter segments of pipe of the same diameter.

1. First connect the 2 T-connectors with two 1.5-meter-long pipes to maintain an upright position.
2. Connect the four short pipes through the T-connectors to maintain stability of the overall frame system.
3. At the top end of the two erect pipes, connect an elbow joint on each of the pipe. Lastly, connect a horizontal pipe of 1.5 meters in the middle of the two elbow joints.
4. Use a velcro tape to secure the Logitech camera onto the horizontal pipe and maintain the camera to be of a fixed straight downward perspective.

Graphical User Interface

The Graphical User Interface (GUI) is used to access control customization over the entire system while the main program is running. It allows for tuning of various parameters, as discussed in the following two sections. The GUI is opened upon running the main code, and will start and stop the main loop, which contains the outer loop control system, computer vision system, and the communication system upon clicking the appropriate button.

In Figure 18.1, the GUI with the various sliders and buttons available can be seen. The start and stop buttons execute and stop the main program, while the sliders adjust specific parameters within the system. Further details about the functionality of the sliders will be discussed in the following sections.

Computer Vision

In order to prevent poor performance of the computer vision system, the camera settings should be set to achieve the best performance possible. The ideal output from the camera will have high saturation and high value, but due to the natural glare produced by overhead lighting, this can at times be difficult to achieve. The effect of imperfect conditions can be mitigated somewhat by tuning the internal camera settings with the official camera settings software.

To tune the camera parameters, download the Logitech camera settings software found at this link: https://support.logitech.com/en_us/product/hd-pro-webcam-c920/downloads Using this

software, navigate to the advanced settings and ensure that the camera settings are set as follows in Table 10.1.

Parameter	Value
Brightness	20%
Contrast	55%
Auto Focus	OFF ~0
Color Intensity	70%
Auto White Balance	OFF and ~3700K

Table ():

Table 10.1: Logitech Camera Settings

Ensure that the camera is at a sufficient height to observe the whole field of play, is facing straight down and is placed in the middle of the field of play. Following this guideline for camera positioning will eliminate many potential issues associated with the recognition of the ball and the robot.

Many parameters used in the computer vision system can be customized by adjusting the sliders found on the GUI. The parameters that can be altered, along with their recommended (default) value and , are explained further in the following table, Table 10.2 [23].

Parameter	Default	Effect
Parameter 1	75	The higher threshold for Canny edge detection- a higher value can eliminate some errant circles due to dirt, floor patterns, etc.
Parameter 2	15	The accumulator threshold for circle center detection- decrease to allow for circles to be more easily recognize, and increase for stricter circle shape recognition.
Minimum Value	125	The minimum HSV color spectrum value a circle must have to be recognized as any given color. Non-robot marking circles are sometimes recognized and can be filtered out by a higher threshold.

Table 10.2: Computer Vision Parameters

In future versions of the GUI, more parameters may be added that will increase the customization of the computer vision system, but with these basic parameters, a set height, and the adjustment of the camera settings, the computer vision system will work as specified. For expert users, other parameters can be adjusted by directly altering the Python code, and include the minimum circle radius, maximum circle radius and various complex color boundaries for circle assignment. All but the last, color boundaries, are defaulted to values that should not require tuning in normal conditions. The color boundaries, should problems be encountered, can be adjusted within the Python script for custom color identification. Colorizer.org, [20], is a useful tool for fine tuning of color boundaries.

Controller

The outer loop control system is started by pressing the start button in the GUI. Also within the GUI are several sliders which adjust parameters for the outer loop PD controller. The user can tune these gain values of the PD controller in order to optimize the robot's speed response and achieve the desired performance for the system. The gain K_p is a scalar multiplier for the error and K_d is a scalar multiplier for the derivative of the error. Setting the gain K_p to 0.65 and K_d to 0.5 ensures that the robot has a stable response.

Code

Github Repository

<https://github.com/MikeTrainor/RoboCupSoccerPlatformDesign>

This product makes use of three different scripts. The first is the python code that contains the vision system and the outer-loop controller, the second is the C++ code for the sending-side KL25Z, and the third is C++ code for the receiving side KL25Z located on the robot. To implement the code for the robot and for the transmitting KL25Z connected to the master computer, first go to <https://os.mbed.com/handbook/mbed-Compiler>, and create an Mbed online account. The Mbed website is used as a compiler to flash the KL25Z microcontroller.

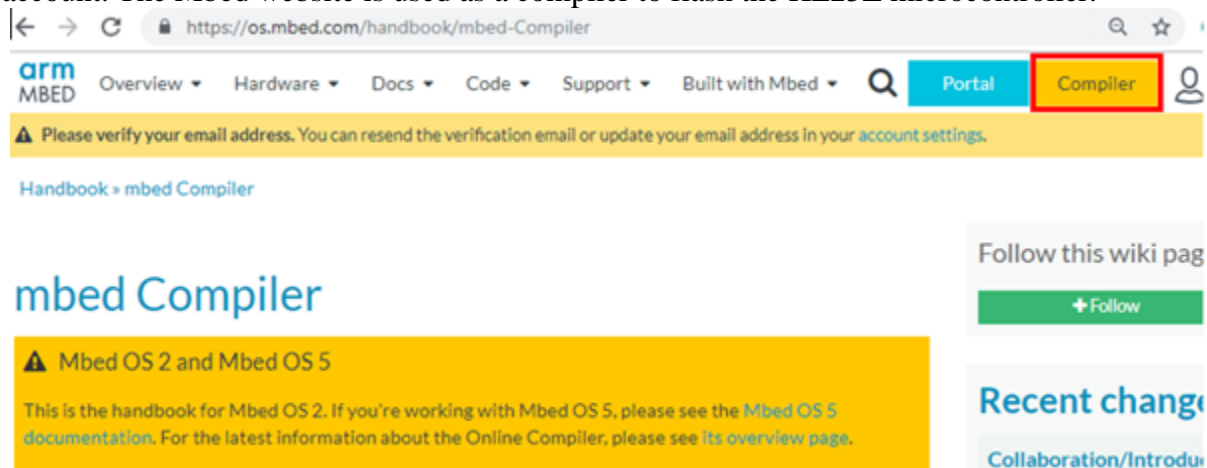


Figure 10.1: Mbed Compiler

Create a new project by clicking on the New tab on the top left of the Mbed compiler.

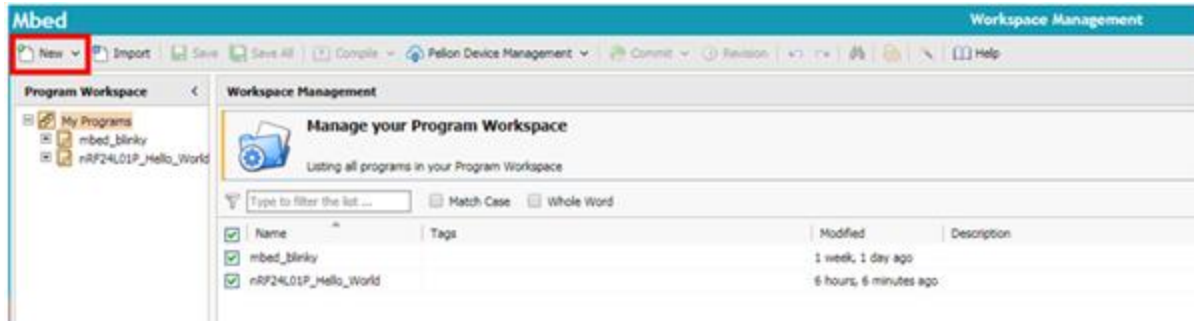


Figure 10.2: Project Creation

Ensure that the selected board is the FRDM-KL25Z as indicated on the top right of the Mbed Compiler. If the selected board is wrong, click on the red box and select the appropriate microcontroller.



Figure 10.3: Board Selection

In the new project that was created, there is a window to the left of the compiler labeled as Program Workspace. This window contains the project folders including the project libraries and main code. Update the libraries to avoid errors, and copy the latest version of the code from GITHUB to the main.cpp file.

Ensure that all the libraries required by the program are imported to the project. The required libraries to run the transmitter side is the Mbed Serial library, which can be found at <https://os.mbed.com/handbook/Serial>

The libraries required to run the code for the robot are listed below:

Mbed QEI <https://os.mbed.com/cookbook/QEI>

Mbed nRF24L01 <https://os.mbed.com/components/nRF24L01/>

Mbed RTOS <https://os.mbed.com/handbook/RTOS>

Mbed Serial <https://os.mbed.com/handbook/Serial>

In the links listed above, import the libraries to the new project. Update all the libraries to avoid errors in the program. The added libraries can be found in the left window labeled as Program Workspace of the Mbed compiler.

To implement the python code for the outer loop controller and vision system, first download Microsoft Visual Studio, and install Python 3.6, or Anaconda 5.2.0 for your system.

To setup your environment for the first time, run VS17 as administrator and create a new project by going to File->New->Project.

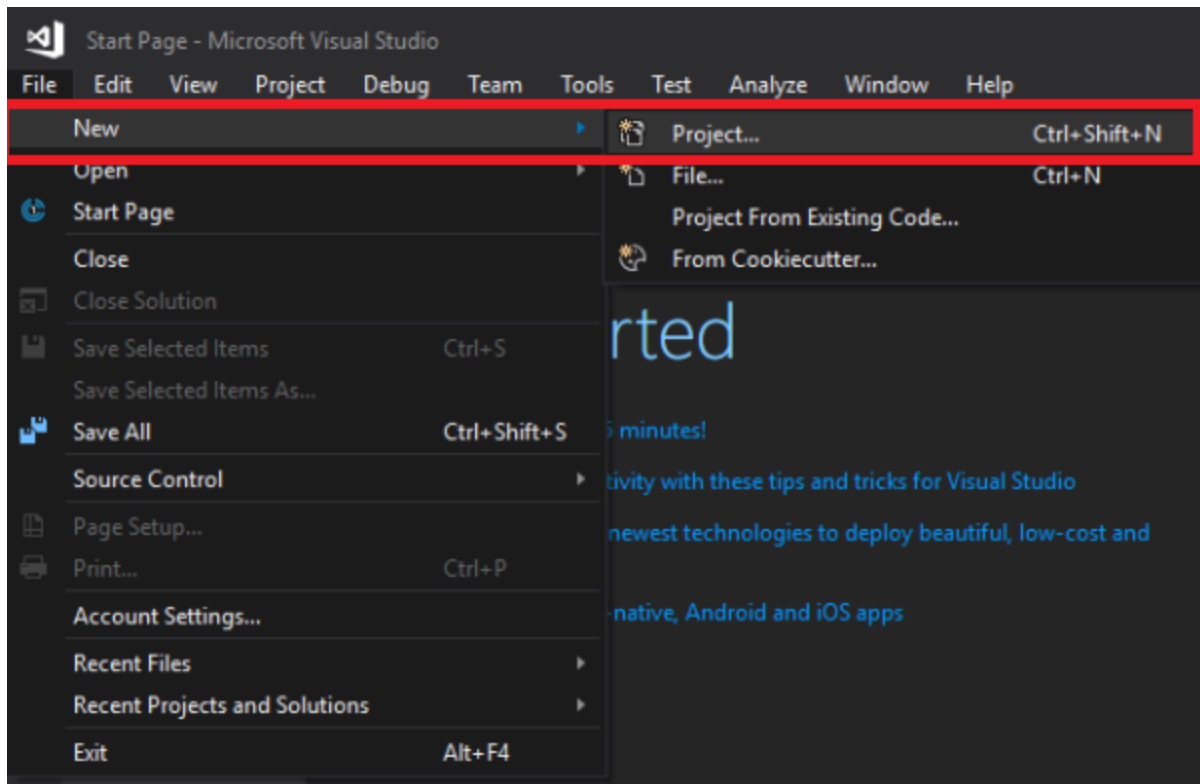


Figure 10.4: New Project

In the New Project window, select Python, in the left column, and then select “Python Application”. Assign a name and a path to your application to complete the creation of your new project

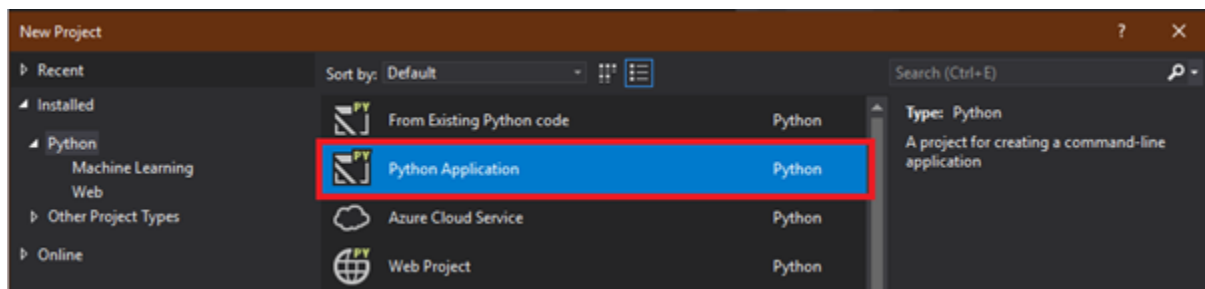


Figure 10.5: Python Application

Next, add the following libraries to your python environment:

- Numpy
- Scipy
- Copy
- math
- Cv2
- Serial
- Time
- Fractions
- Sys
- PyQt5

To install these libraries, open up the Python Environments located on the right hand side of the screen ,or by going to View -> Python -> Python Environments, in the VS17 drop down menu. Select Anaconda 5.2.0, and ensure that this is your default python environment, by clicking on “Make this my default python environment”. In the drop down menu select Packages (Conda). The search bar can then be used to search for the packages. If the package does not show up, “Run command: conda install” may be clicked instead. An aAlternatively, to this, is to install the libraries may be installed through Packages (Pypi).

Note that any libraries do not have to be installed in both Pypi and Conda to be accessible by your python environment.

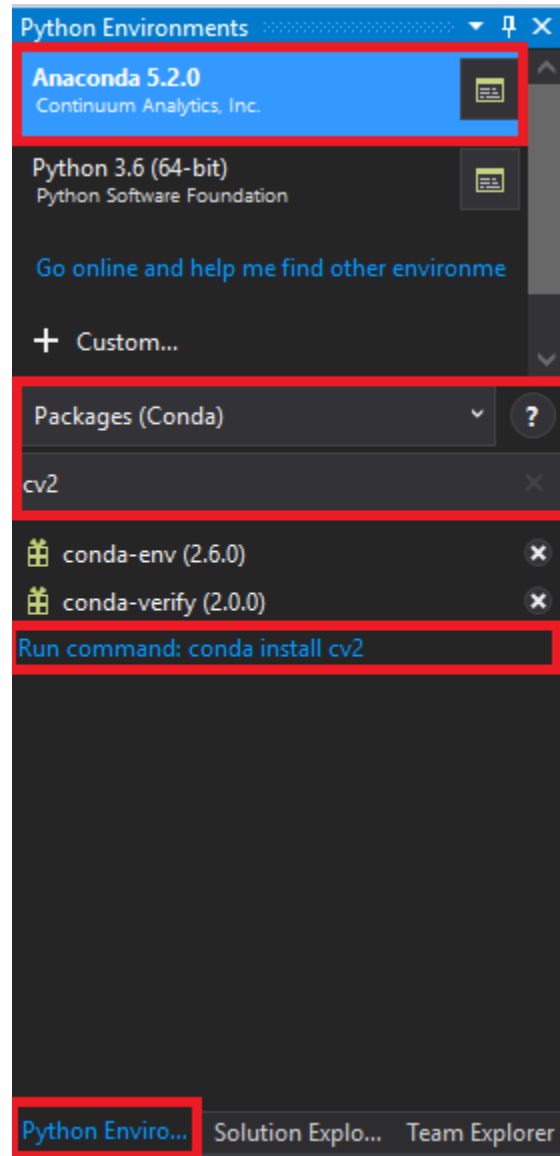


Figure 10.6: Python Environments

Next the code can be copied from the Github repository and pasted into the python document. To run the code, press the Attachment/Start button.

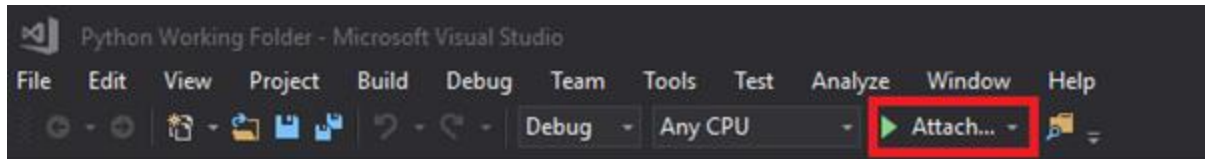


Figure 10.7: Start Program

Robot

Using Table 18.2, complete each of the necessary connections. Double check the connections before powering up the robot as making a wrong connection may damage/destroy components.

Before using the robot, ensure that the battery is charged by checking the battery monitor, once this is complete the robot may safely be used. Turn on the robot by pressing the rocker switch located at the back of the robot. The robot is now ready for operation.

Pin connections:

Robot		
Pin	Application	Connection
Battery	Robot power source	Positive terminal (blue wire with rocker switch), Negative terminal (green wire)
Battery Voltage Level Tester	Battery level monitoring	Align the negative balancer port from the battery with the negative port of the battery voltage level tester pin. The reading indicate battery voltage level per cell.
Vin	Power for microcontroller	5V regulator
3.3V	Power for NRF24L01	NRF24L01 Vcc
GND	Voltage reference	NRF24L01, encoder 1, encoder 2, battery, 5V regulator
PTE22	PWM motor 2	PWM 2
PTE20	PWM motor 1	PWM 1

PTE4	Direction select motor 1	DIR 1
PTE5	Direction select motor 2	DIR 2
PTD4	Motor 1 encoder channel 2	Motor 1 CH 2
PTA12	Motor 1 encoder channel 1	Motor 1 CH 1
PTA4	Motor 2 encoder channel 2	Motor 2 CH 2
PTA5	Motor 2 encoder channel 1	Motor 2 CH 1
PTD0	SPI chip select	NRF24L01 CSN
PTD1	SPI clock	NRF24L01 SCK
PTD2	SPI slave data input	NRF24L01 MOSI
PTD3	SPI slave data output	NRF24L01 MISO
PTD4	Kick command	TIP102 Base
PTE0	NRF24L01 chip enable	NRF24L01 CE

PTE1	NRF24L01 maskable interrupt	NRF24L01 IRQ
KL25Z Transmitter		
Pin	Application	Connection
PTD0	SPI chip select	NRF24L01 CSN
PTD1	SPI clock	NRF24L01 SCK
PTD2	SPI slave data input	NRF24L01 MOSI
PTD3	SPI slave data output	NRF24L01 MISO
PTE0	NRF24L01 chip enable	NRF24L01 CE
PTD5	NRF24L01 maskable interrupt	NRF24L01 IRQ
3v3	Power for NRF24L01	NRF24L01 Vcc
GND	Voltage reference	NRF24L01 GND

Table 10.3: Pin Connections

Operation

The battery is designed for the robot to be able to play a 10 minute game. Game time should be limited to within this time period. During operation, the robot will attempt to kick the ball into the center of the net. After each goal, the ball must be reset by a referee to your chosen location. This location must not be along the outer border as this would cause the robot to go out of frame when trying to retrieve the ball.

Should the ball go out of frame, it is necessary to manually retrieve the ball and place it on the field. In this sense, the user is effectively the referee for the game.

1. Run the python code.
2. Turn on the rocker switch located on the robot.
3. Observe the fact that the robot approaches the ball and kicks the ball.

Maintenance

Maintenance for the robot consists of recharging the battery when the battery monitor begins to beep. This can be done by disconnecting the battery from the robot and charging it with a 3s LiPo Balancer Charger.

A tutorial on how to use the LiPo Balancer Charger to charge up the battery in the robot can be found here: <https://www.youtube.com/watch?v=RPzJOKoHhhQ&t=962s>

1. Connect the charging port from the battery with the 3s charging port of the IMAX B6 charger.
2. Use banana plugs to connect the battery with the charger.
3. Plug the adapter of the balancer charger into a wall socket and hear the beep coming from the charger.
4. Change settings on the balancer charger to Lipo ->Charge -> 0.7A ->11.1V(3S), by pressing and releasing left, right and enter key in a small time interval. "Left" and "right" is meant to increase or decrease values under a fixed setting while "Enter" is to change to the next setting parameter. Once every setting is made, user can press and hold the enter key for roughly 3s to let the balancer charger check that no mistake has been made.
5. Hit the confirm button when suggested by the balancer charger.

To disassemble the robot, four Philips-head screws must be removed from the top of the robot. Following this, remove the lid by pulling it straight up.

Robot Assembly

To build the robot frame, download the .stl models of the robot frame, the robot top of the frame, the wheels, the kicking mechanism, and the solenoid holder from the GitHub repository provided in the Github section. Using a 3D printer, print the models.

Complete the PCB implementation and wiring of all the components, secure the solenoid to the solenoid holder, and then secure the kicking mechanism using a nail to the solenoid.

Superimpose the PCB on top of the kicking mechanism, and ensure that all wires are not interfering with the kicking mechanism.

Throughout this process, use standoff screws to create spacing between the components. These screws are used to secure the robot frame and the top of the robot frame together.

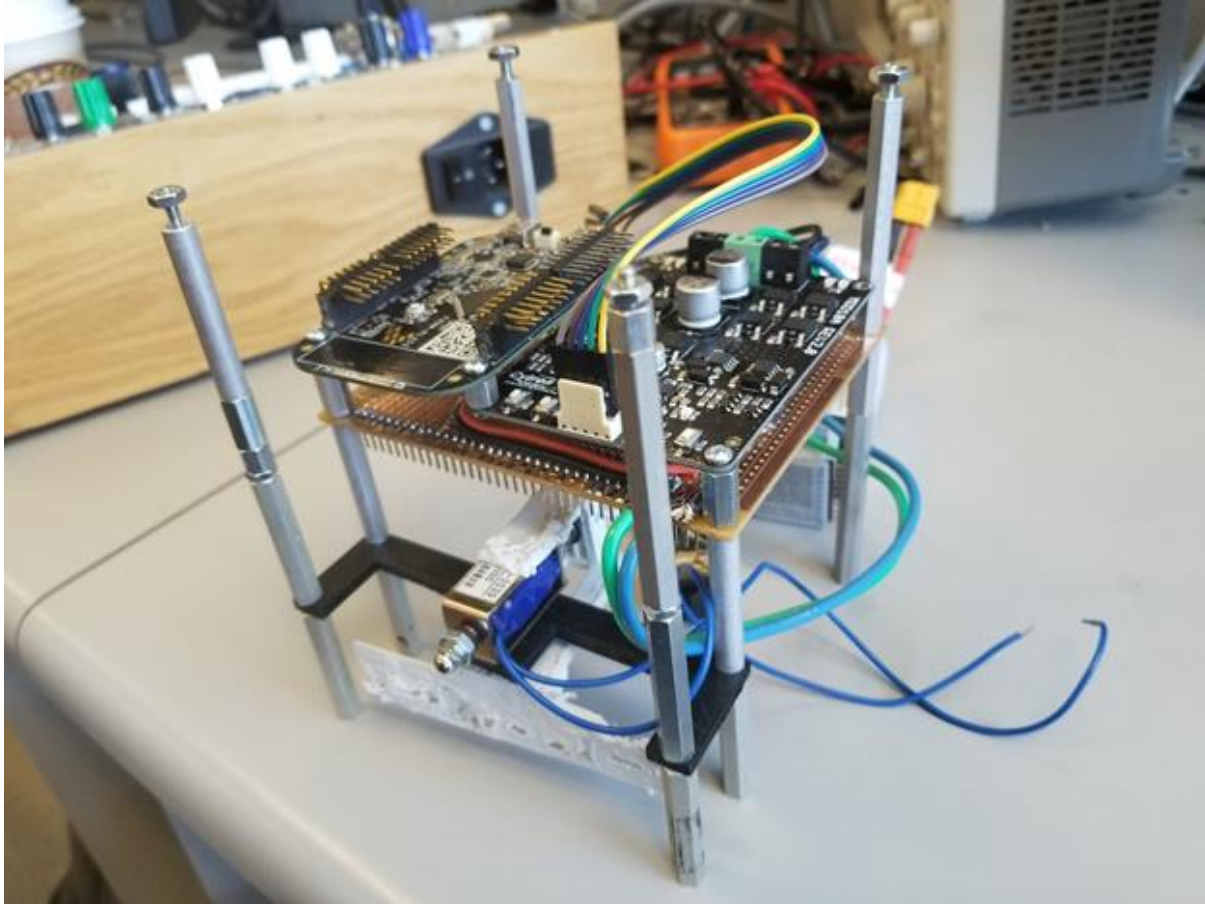


Figure 10.8: Robot Interior Design

Secure the 2 motors to their respective placement in the robot frame. Each motor require 6 screws.

Note that for the motors to fit within the restriction defined by Robocup, the motor encoders caps to have the necessary space to mount both motors in the frame. Once completed, ensure that the two encoders are separated far enough to not cause physical interference with one another.



Figure 10.9: Motor Location

At this point you can screw in the 6 screws used to secure the PCB and kicking mechanism with the standoff screws.

The marble provided acts as a spherical wheel used to stabilize the robot. Secure it to the bottom of the frame using 3 screws as follows.



Figure 10.10: Bottom View of the Robot Showing the Caster Wheel

Secure the PCB, kicking mechanism and the lower part of the frame to the top of the frame using 4 nuts. Attach a robot team and player ID according to the RoboCup regulations.

The robot is now ready to kick the ball!



Figure 10.11: Complete Design of the Robot

Troubleshooting Guide

This troubleshooting guide will help explain common issues that may arise, identify the potential cause of the issue, and how to fix it.

Issue: The robot does not move or briefly moves when the KL25Z on the robot is reset.

Cause: The buffer containing the data being sent was not empty prior to obtaining new data, resulting in shifted bits being sent.

Fix: With the receiving KL25Z running, the sending KL25Z must be reset by pressing the reset button located between the two USB ports as shown in Figure 18.8.

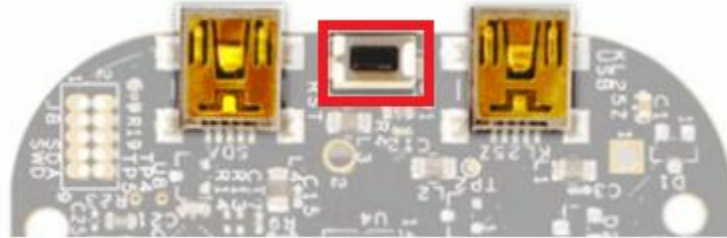


Figure 10.12: KL25Z Reset Button

Next the Python code must be executed. This will ensure that the RF is sending the correct information to the receiving side.

Issue: The wheels are spinning the opposite direction as intended.

Cause: Incorrect wiring

Fix: Consult Table 18.2: Robot Pin Connections to ensure proper connection of the wires.

Issue: The robot keeps rotating quickly back and forth, or in place.

Cause: The overshoot is too high so that the robot goes beyond the desired angle

Fix: Using the GUI, lower the Kp and Kd gain values for the controller

Issue: The sending side KL25Z has an error file after the code has been flashed to the board.

Cause: This is a bug in Windows 8.1 and Windows 10.

Potential Fix: Follow the instructions shown on the following page.

<https://superuser.com/questions/1199823/how-to-prevent-creation-of-system-volume-information-folder-in-windows-10-for>

This fix may not work for all computers.

Issue: The vision system is miscalculating the angle of the robot

Cause: All of the circles on the robot are not being recognized at once. This may be due to external interference (such as poor lighting), noise or objects being recognized as circles.

Fix: Ensure that the field, robot and ball are the only objects within the camera frame as well as the playing surface being flat and free of creases.

If this does not solve the problem, it may be that the parameters for the Circle Hough Transform are poorly tuned. Attempt to tune the parameters to recognize only the correct circles- those on the robot and the ball. Details of the effect of each parameter can be seen in the Computer Vision section of the user guide.

Issue: The vision system is not recognizing any of the circles on the robot

Cause: The height of the camera is too low or too high, or the parameter settings for circle recognition are too strict.

Fix: Adjust manually the height of the camera, and if that solves nothing, adjust the parameters to be more relaxed, allowing circles to be better recognized.

Issue: The vision system is detecting multiple robots or balls.

Cause: The background may be excessively noisy, or the parameters may be improperly tuned.

Fix: Tune the parameters in the GUI, until you see the ball and the robot being detected correctly. More detailed discussion on the effect of each parameter can be seen in the Computer Vision section of the user guide.

Issue: The vision system recognizes the robot well, but the ball is not recognized.

Cause: The tuning of the computer vision system in the GUI may be good, but the tuning of the camera settings was poorly done, and the high glare does not allow for the shiny orange ball to be recognized correctly.

Fix: Adjust the parameters in the camera settings application for the Logitech webcam to reduce glare, increase saturation, and increase value in the robot markings and ball as much as possible. Suggested settings can be found in the user guide.

Issue: The battery alarm is beeping

Cause: The battery is discharged

Fix: Disconnect the battery from the robot, and recharge the battery by connecting it to a 3s Li-Po battery balancer.

Issue: The transmitting side KL25Z displays a solid yellow LED

Cause: Wires have become disconnected between the KL25Z microcontroller and the NRF24L01 or have been connected incorrectly.

Fix: Consult with Table 18.2 to ensure proper reconnection of the wires.

Issue: A circle is being detected on the playing surface without any object being there

Cause: A circle may be detected if dirt/dust is on the surface in the shape of a circle

Fix: Remove the dust or dirt from the playing field

Issue: Cannot install the required python libraries

Cause: Administrator rights are required to download libraries

Fix: Microsoft Visual Studio must be run as administrator by right clicking the icon and selecting "Run as administrator"