

pybd_gui Overview

pybd_gui is designed to help students in EGR 345 lab auto-generate Arduino or Raspberry Pi code for performing real-time experiments on dynamic systems. These experiments may include feedback control.

In order to auto-generate the code, the software takes a template file and inserts code blocks at certain points. These code blocks are based on the block diagram system that students create using the gui. (A GUI is a graphical user-interface.)

Code insertion points

If you search through the template file, you will find the following points where pybd_gui inserts code:

//bdsysinitcode

This is the code that initializes or creates the blocks in the model.

//bdsyswelcomecode

This is a spot where a welcome message can be printed that would help verify that the correct code is being run.

//bdsyssetupcode

The main purpose of the setup code is to specify the inputs for each block. Specifying that certain pins are outputs would also be done here. This code is placed inside the `setup` function for an Arduino.

//bdsysmenucode

The menu code is mainly for allowing the user to set certain block parameters before running a test. An example of this would be allowing the user to change `kp` and `kd` for PD tuning.

//bdsysloopcode

This is the “main” code that runs the control logic for each time step of the experiment. This code is inside the `loop` function of an Arduino.

//bdsysprintcode

The print code handles printing data to the serial monitor for Arduino or saving data to a csv file if using Arduino as the main device. When building the block diagram system model, you have to set the print blocks to determine which blocks outputs are printed.

Model Creation

A block diagram system is made up of blocks, sensors, and actuators. The blocks may have inputs, which can be other blocks or sensors. In order to create a system model, you need to perform the following steps:

- create sensors and actuators
- create blocks
- specify block inputs
- place blocks
- specify the print blocks

Note that actuators and sensors must be created before a plant block can be create. Plant is a controls word that means “the thing we are trying to control”. So, a plant can be a circuits, a motor, a robot, an autonomous vehicle, a manufacturing process, Any dynamic system can be a plant.

Because you cannot manually move or connect the blocks, you have to tell the software how to place them on the block diagram and what the inputs for each block are.

Block Placement

The best practice is to have one block that is placed with absolute coordinates. The rest of the blocks can then be placed relative to other blocks. When using the relative method to place a block, the block you are referencing must already be place. So, start with your absolute block and move outward from there. That means it is probably best if the absolute block is the first input and all the other blocks are place in one direction from there.

You can press the draw button whenever you want to see the current state of your system model. Only blocks that have been placed will be drawn.

Print Blocks

If you had a very complicated model, it would be confusing to create a csv file with many columns. So, the software allows you to choose which blocks have their output printed. If you are using an Arduino as your main or only control device, the output from the print blocks will print to the serial monitor. If you are using a Raspberry Pi as the main device (probably with i2c between the Raspberry Pi and one or more Arduinos), then the output may be saved directly to a csv file.

Code Generation

As discussed above, the software generates the code for real-time experiments by inserting code for each block at various points in the code template file. So, a code template must be specified. Before auto-generating the code, you must tell the software where to find the template and where to put the output. If you are generating Arduino code, you specify the folder that will contain the *.ino file. If you are generating Raspberry Pi code, you specify the name and location for the *.c file.

Once you have specified the template file path and the output destination, you are ready to generate the code by choosing either **Generate Arduino Code** or **Generate RPi Code** from the menu (generating RPi code will come later in the semester).