# Introduction

The following guide has been designed to instruct the user on how to configure the software (scripts, functions, programs, etc.) that controls that control the hardware (electrical components, modules, sensors, etc.) that compose the prototype.

# Objective

The following guide will instruct the user on how to establish the slave-master interaction between Arduino-controlled prototypes and Raspberry Pi-powered control systems [fig].

# Materials

The following components are required for the complete use of this guide. For more details and a complete list of components comprising the prototype, the user may access the BOM.

1. Arduino Pro Mini 328 3.3V 8MHz
2. BlueSMiRF Bluetooth Modem (RN42)

# Configuration of Bluetooth antenna

All of the prototypes developed for this project transmit or relay sensor data wirelessly, via a Bluetooth antenna. This section of the guide will support the user through the processes of pairing and connecting the Bluetooth antenna to a control computer. For the extent of this guide, the control computer has been assumed to run a Windows operating system.

## Pairing with control system

1. Power the Bluetooth antenna
   1. Red flashing light on the breakout board will indicated power on but disconnected
2. On the control computer, add a Bluetooth device
   1. The user may search “add Bluetooth device” under the start menu
3. To identify the Bluetooth antenna within the “Add a device” window, compare the IDs listed to that printed on the antenna
4. Select the Bluetooth antenna and chose the second pairing option “Enter the device’s pairing code”
   1. Enter the pairing code “1234”
   2. Most antennas have been encrypted with simple pairing codes such as “0000” or “1234”, for more information consult the antennas user guide [ref]
5. Wait until the control computer installs the COM ports associated with the antenna
   1. Note that serial communication may not be establish directly through this interface. In other words, a successful connection cannot be demonstrated at this point
   2. If the installation completes successfully, the user may assume that a connection is feasible
6. In order to verify the COM port number for the antenna, the user may right-click on the Bluetooth device (within the list of paired devices), select “Properties” and then look under the “Hardware” tab

## Configuring the module

Bluetooth antennas or modules can be configured using command line prompts, terminals, or even the Arduino itself. Depending on the module, configuration may encompass the ability of changing the module’s name, address, pairing code, and more relevant communication features such as parity and baud rate.

This guide will show the Arduino method in order to reduce the amount of hardware and software needed. For more information on alternative methods for Windows computers, the user may consult the SparkFun BlueSMiRF hookup guide [ref].

1. Open the Arduino IDE
2. Without writing any code, select the COM port used by the antenna (Tools > Port)
   1. Ensure the antenna is powered on and disconnected (flashing red light)
3. Open the Serial Monitor (Tools > Serial Monitor or Ctrl+Shift+M)
4. Type the command “$$$” and press {Send} or [Enter]
   1. Ensure that “No line ending” has been chosen as the communication option (bottom-right section of Serial Monitor window)
   2. The response “CMD” should be sent by the module
5. Change the communication option from “No line ending” to “Newline” (bottom-right section of Serial Monitor window)
6. Type the command “D” and press {Send} or [Enter]
   1. This command will prompt the module for the configuration information or settings
7. Type the command “SU,57k” and press {Send} or [Enter]
   1. This command will change the baud rate (default either at 9600 or 115200) to 57600
   2. This change will allow the seamless communication between the Arduino (using StandardFirmata) and the Processing algorithms within the control system

# Configure Arduino

All of the prototypes developed for this project use Arduino modules to acquire and convert signals from selected sensors. While Arduinos can be programmed to process the acquire data, the project’s prototypes used a type of “pass-through” script that allows for such control and/or processing to be performed by the control system. This script, known as StandardFirmata, was developed by Arduino and Processing community members for a hardware configuration similar to that of our project.