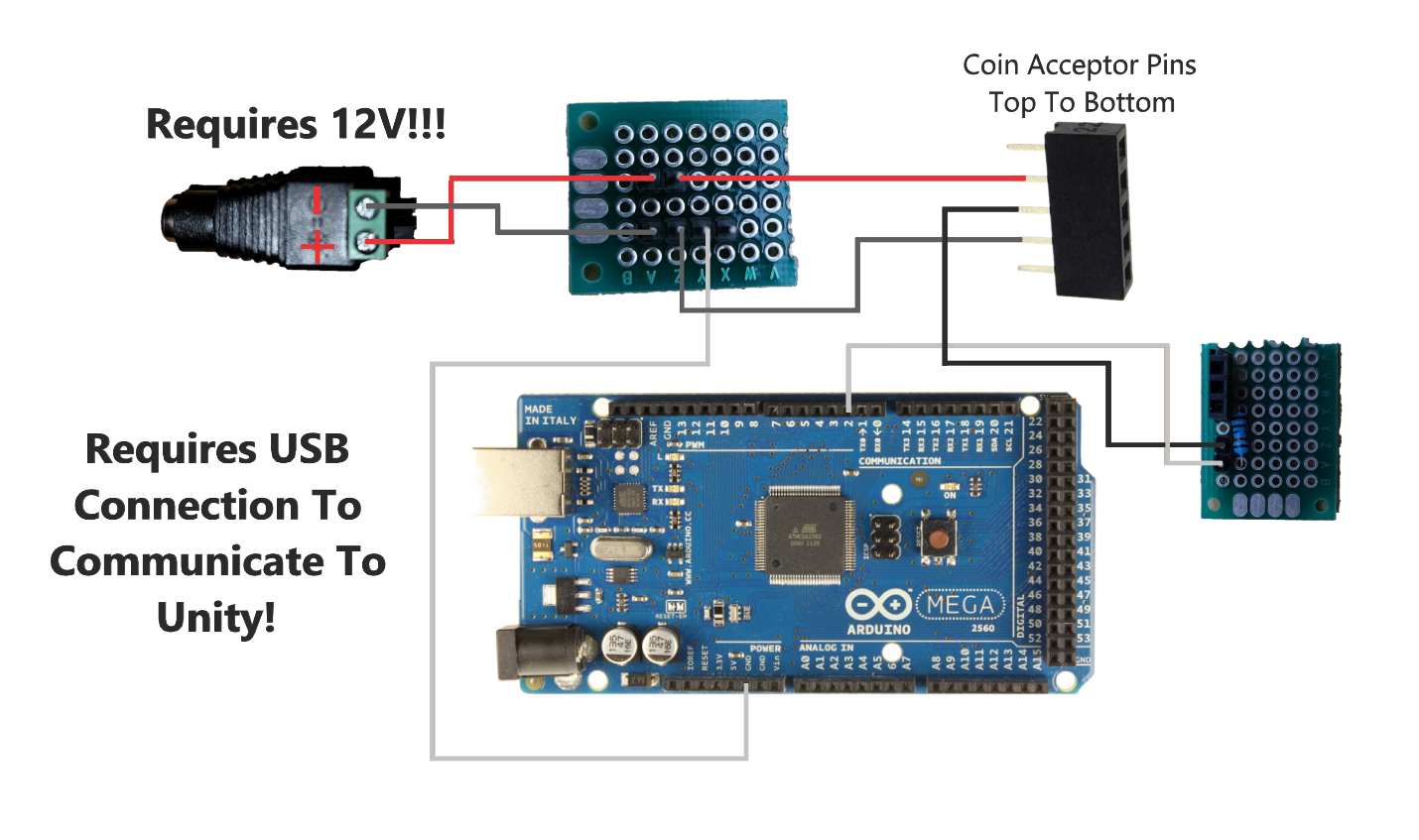
**Coin Acceptor Documentation**

**How It works**



Above is a wiring diagram showing how to properly wire up the coin acceptor. Let’s start by looking where it says Coin Acceptor Pins. This part of diagram is used to represent the back of the coin acceptor where there are 5 header pins. The second header pin from the top is the 12V connection. The middle header pin is where the impulse is sent from when a coin is inserted, and the fourth header pin is for ground. Please note the first and last header pins are not used.

If you look at the part of the diagram where it says Requires 12V you’ll notice a red wire going from the 12V plug to a custom PCB component. Will call this PCB1 for now. PCB1 has two parts to it. It has 2 header pins that are bridged at the top and then 4 header pins at the bottom that are also bridged. When you hear the term bridged if you look at the bottom of PCB1 you can see that each set of header pins is soldered together allowing currents to pass through them. The top 2 header pins where you see the red wires connecting start from the 12V connection using a M-F DuPont wire to connect to the first header pin and then a F-F DuPont wire goes to the Coin acceptor giving the coin acceptor power when plugged in. The bottom four header pins are all ground connections. The first gray wire goes to the negative connection of the 12V plug the second to the Arduino(Elegoo) board ground and the third to the coin acceptor ground. The last header pin is just extra and has no use.

If you look over at the right of the diagram you will see a second PCB component which we will refer to as PCB2. This PCB component has two bridged male header pins and three bridged female header pins as well as a resistor. The female header pins and resistor were originally intended for a display, but my best guess is the soldering was not done properly on this PCB component and there for it does not work. However, the two male header pins do. The top header pin where you see a black F-F DuPont wire runs from PCB2 to the middle male header pin of the coin acceptor. This header pin is where the coin acceptor sends impulses from when a coin is inserted. The impulses travel from the coin acceptor to PCB2 via the top DuPont wire. From this point there is a second F-M DuPont wire connected to the bottom female header pin that runs from PCB2 to the Arduino(Elegoo) board and is connected to female header pin 2 on said board. That essentially sums up the wiring and how it all functions properly. I will also include a line below where I based this design off of.

**This link also shows how to program the coin acceptor to accept more coins!!! DO NOT DO THIS UNLESS YOU FULLY UNDERSTAND THE CODE!!!**

**Original Design:** [**https://www.youtube.com/watch?v=sfE7yqtd8TA**](https://www.youtube.com/watch?v=sfE7yqtd8TA)

**Also please note as far as hooking this up to a computer goes you need to have the Arduino hooked up via a USB port to the computer and the 12V plug is needed to power the coin acceptor.**

**Code For Unity And Explanations**

To use this script in unity simply create an empty game object and drag the script from the menu on the game object.

The only thing worth noting in this import section is System.IO.Ports which is used to communicate to the micro controller plugged into the PC.Text

Description automatically generated

Line 17 you will notice by the class name the words MonoBehavior this part is needed for unity to recognize this as a script. Line 19 is just an integer used to track the amount of coins inserted. Line 22 is a Unity UI Text component. It is set to public so that after you have applied the script to a game object you can simply create a text object and drag it to the menu options of the script.

SerialPort sp = new SerialPort(“COM6”, 9600) is the most important part of this file. COM6 specifies what port the coin acceptor exists on. This port is dynamic and changes often. Every time you disconnect or restart the computer you run the risk of changing it. Currently this is hardcoded in the code. It might be ideal to create a new script that simply asks for the correct COM and then allows the program to dynamically change said COM. If you need to look up the current COM being used by the coin acceptor the easy way to do this on windows is under Device Manager and then simply look under Ports (COM & LPT). The port is usually labeled as a USB Serial Device and easily noticeable.Text

Description automatically generatedGraphical user interface, text, application, email

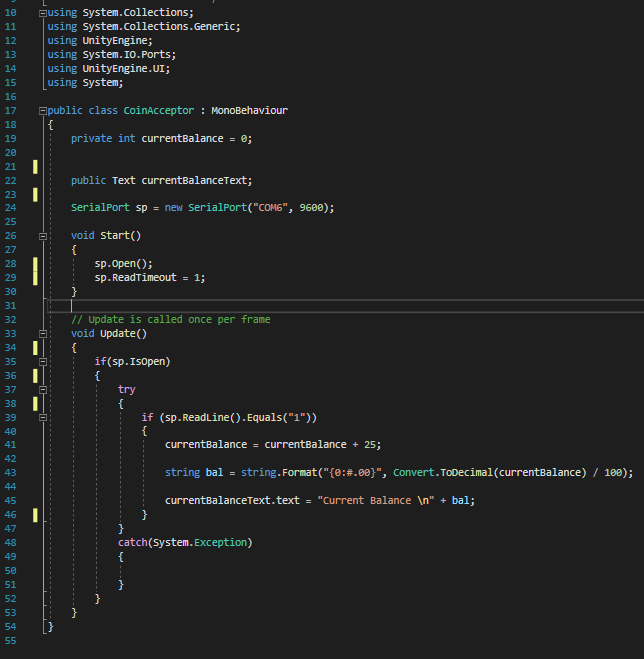
Description automatically generated

Void start is the first method that is called on the start of the object. sp.Open() simply opens the line of communication. Important note for this though is if you also have a serial monitor open under the Arduino IDE it will cause errors related to ACCESS DENIED. If you see this error it usually means something else is already accessing the port.Text

Description automatically generated

The method Update runs every frame. Essentially while the script is active this is a never ending loop. sp.isOpen simply checks if the port was opened properly.Text

Description automatically generated

Line 39 is where it looks to see if a coin was inserted. If a coin was inserted the microcontroller is programmed to output “1” after which the unity code which is monitoring the port for communications will see this and read it in causing the currentBalance to increment and modifying the Text view.

This last part just a catch for possible exceptions. It might not be a bad idea modify this to handle exceptions as you discover what’s possible.

Text

Description automatically generated

**Code For Arduino And Explanations**

This code is the code that needs to be uploaded to the micro controller so it can communicate to the PC via the serial port when it receives pulses from the coin acceptor.

Text

Description automatically generated

The setup method is pretty straight forward . Line 11 just opens up the lines of communication with the PC. Line 14 is essentially catching when an impulse comes in and fires the incomingImpuls method.Text

Description automatically generated

This portion of the code is used to count the impulses. The number of impulses is based on how the coin acceptor is programmed to recognize coins.

Text

Description automatically generated

Currently the coin acceptor is only able to recognize one coin which is a quarter. When this coin is recognized it will then output the “1” to the serial port and then if the Unity Script is running it will catch that output via the sp.ReadLine() method in use above.