

# Recy-Claw User Guide

## Description

The Recy-Claw is a claw machine type game with point scoring music files. The claw game functionality is through the game control board. The score display and music is controlled by a Raspberry Pi. Audio Files are located on a flash drive connected to the Raspberry Pi. See sections below for information on audio tracks. The game screen and music takes ~30 seconds to fully boot. Once powered on, the music and screen should start automatically. This will play a welcome audio and display a message on the matrix. Once the start button is pressed, the matrix and music will change to game mode. The user will have a predetermined number of tries, as defined in setup instructions.

## Audio Files

Audio files are located on the USB Drive. The files should be in the main directory of the USB Drive. The naming scheme is “TRACK000.wav” Where “000” is the track number. Audio files must be “wav” format with a frequency of 44100Hz, mono track, and 16 bit encoding. The audio files will be played on loop, so ensure file will sound good in a loop. Score and End audio files are not on loop.

## Track Numbering

TRACK001.wav: Welcome Music, Idle Machine  
TRACK002.wav: Game Music, In-Game Music  
TRACK003.wav: Score, Normal Score Point  
TRACK004.wav: Score, Bonus Score Point  
TRACK005.wav: End Music, Game over

## Code Files

The setup.py script has all the imports and most of the functions. Use this file to set the Beam Break Sensor(s) input Pin, start button input pin, claw machine count pin. Changes to the matrix (screen) output can be changed in the “run\_time.py” script. For troubleshooting, there is a “diagnostics.py” script. Use caution when manipulating source code. See Technical Documentation for more details.

## TROUBLE SHOOTING

### No Display on Power Up

Does the Raspberry Pi have Power? Does the LED matrix have Power?

Check the “cronlog”, log into RPi and type “cd ~/logs”, type “cat cronlogs” to view error messages.

**No Audio**

Check power and volume control for audio amplifier. Ensure the headphone output from the Rpi is properly connected to the audio amplifier.

**Restart Program (after changing code)**

Log into RPi

Go to folder: “cd ~/Documents/clawMachine/src”

“sudo python3 run\_time.py”

Assuming any error have been correct, a reboot will also restart the program.

**Run Diagnostic Script**

Go to folder: “cd ~/Documents/clawMachine/src”

“sudo python3 diagnostic.py”

error messages will appear on screen / command line

**Error Messages**

Messages will print to the command line and the display (if available)

Message Format is: Error #, File.Line.#

Example message-> ERR: 10 S.L.8 ==> Error 10 in setup file line 8

ERR 10 - Import module error – Check setup file

ERR 12 - Name Errors – Check Import Module declarations and usage

ERR 20 - Audio File Error – Check Name and Format

# Technical Documentation

## Connecting and Editing Files / Code on the Raspberry Pi

Code Files located at <https://github.com/arcanaworkshop>

### Connecting to the Raspberry Pi (RPI)

#### Windows

Use PuTTY, found at <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>. Download and install program

Connect Ethernet cord from laptop to RPi

Open PuTTY, under “Session” use Host Name “raspberrypi.local”

Use credentials below

To share internet with RPi, ensure internet sharing is active. Go to wifi settings and select “Allow other network users to connect ...”

#### Mac / Linux

Connect Ethernet cord from laptop to RPi

Open Terminal

type: “ssh pi@raspberrypi.local”, see Credentials for password

To share internet with RPi, ensure internet sharing is active. Go to sharing settings and select “Internet Sharing”

#### Credentials

user: pi

password: arcana

If using sudo commands, use the same password. The password will not appear while typing.

### Updating Code

Once connected to the Pi, you’ll be using command line instructions. Go to the source code folder and use ‘git’ commands to pull the latest version of code. Before updating code, be sure to kill the currently running program (see below). To edit files on the RPi, you can use “nano ‘fileName’”, where ‘fileName’ is the file you wish to edit. It is advised to change files remotely using git repository, then pulling changes to the Rpi.

**To kill the existing program** (required to update and re-run the program)

In the command line:

“top”

Look for “python”, should be near the top of the list. Take Note of the “PID” value.

“CTL-C” to exit the list of information

“sudo kill PID”, where PID is the value you noted

Now you can go to the source files as indicated above update and re-run the “run\_time.py” script.

### **Updating code from GitHub repository**

cd Documents/clawMachine/src

git pull

To rerun program, type: sudo python3 run\_time.py

NOTE: To see which branch you’re on, type: “git branch”

### **Git Commands (To pull / push code changes to remote files)**

A few important Git commands to use when logged into the RPi:

For getting files from the repository:

“git pull” – get latest information from repo

“git fetch origin [remoteBranch]:[localBranch]” – where [remoteBranch] is a new branch that you need to pull. [localBranch] will be the same name

“git branch” – lists the branches on the repository. If you tried adding a branch, make sure it shows up in this list.

“git checkout [branch]” – where [branch] is the branch you want to switch to

### **Adding files to the repository**

“git add [files]” – where [files] are local files you want to add to the repo

“git commit -m ”message“” – where “message” is user defined

“git push” will push the committed files to the current repository