Walkmellotron: Cassette Player Mods

Created by John Park

https://learn.adafruit.com/walkmellotron

Last updated on 2023-08-29 04:38:58 PM EDT
## Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>5</td>
</tr>
<tr>
<td><strong>Walkmellotron MIDI Controlled Tape Player</strong></td>
<td>6</td>
</tr>
<tr>
<td>• Parts</td>
<td></td>
</tr>
<tr>
<td>• Cassette Player</td>
<td></td>
</tr>
<tr>
<td><strong>Build the Walkmellotron</strong></td>
<td>9</td>
</tr>
<tr>
<td>• Tape Player Mod</td>
<td></td>
</tr>
<tr>
<td>• Solder to VS+</td>
<td></td>
</tr>
<tr>
<td>• Solder to GND</td>
<td></td>
</tr>
<tr>
<td>• Wire Run</td>
<td></td>
</tr>
<tr>
<td>• Wire to DAC</td>
<td></td>
</tr>
<tr>
<td>• DAC to QT Py, come in QT Py</td>
<td></td>
</tr>
<tr>
<td>• Affix!</td>
<td></td>
</tr>
<tr>
<td><strong>CircuitPython</strong></td>
<td>14</td>
</tr>
<tr>
<td>• CircuitPython Quickstart</td>
<td></td>
</tr>
<tr>
<td>• Safe Mode</td>
<td></td>
</tr>
<tr>
<td>• Flash Resetting UF2</td>
<td></td>
</tr>
<tr>
<td><strong>Code the Walkmellotron</strong></td>
<td>18</td>
</tr>
<tr>
<td>• Text Editor</td>
<td></td>
</tr>
<tr>
<td>• Download the Project Bundle</td>
<td></td>
</tr>
<tr>
<td><strong>Jam on the Walkmellotron</strong></td>
<td>19</td>
</tr>
<tr>
<td>• Source</td>
<td></td>
</tr>
<tr>
<td>• Playing the Walkmellotron</td>
<td></td>
</tr>
<tr>
<td><strong>Motor Controlled Cassette Player</strong></td>
<td>23</td>
</tr>
<tr>
<td>• Electronic Parts</td>
<td></td>
</tr>
<tr>
<td>• Cassette Gear</td>
<td></td>
</tr>
<tr>
<td><strong>Circuit Diagram</strong></td>
<td>27</td>
</tr>
<tr>
<td><strong>3D Printing</strong></td>
<td>28</td>
</tr>
<tr>
<td><strong>Code the Motor Controlled Cassette Player</strong></td>
<td>30</td>
</tr>
<tr>
<td>• Upload the Code and Libraries to the QT Py RP2040</td>
<td></td>
</tr>
<tr>
<td>• How the CircuitPython Code Works</td>
<td></td>
</tr>
<tr>
<td><strong>Soldering</strong></td>
<td>33</td>
</tr>
<tr>
<td>• Components</td>
<td></td>
</tr>
<tr>
<td>• Socket Headers</td>
<td></td>
</tr>
<tr>
<td>• DIP-8 Socket</td>
<td></td>
</tr>
<tr>
<td>• Terminal Blocks</td>
<td></td>
</tr>
<tr>
<td>• Button</td>
<td></td>
</tr>
<tr>
<td>• Switch</td>
<td></td>
</tr>
<tr>
<td>• Potentiometer</td>
<td></td>
</tr>
<tr>
<td>• Wiring</td>
<td></td>
</tr>
<tr>
<td>• GND Signals</td>
<td></td>
</tr>
<tr>
<td>• 3.3V Signals</td>
<td></td>
</tr>
</tbody>
</table>
- 5V Signals
- Button Input
- Potentiometer Wiper
- PWM Connections
- Motor Connections
- Cassette Player Wiring
- Motor Wiring
- Power Connections

Assembly

Make a Cassette Tape Loop

Usage
Overview

Portable cassette tape players aren't just the best way to listen to music on the go, they also can be turned into musical instruments in their own right! By adjusting playback speed of the tape motor, you can increase or decrease the pitch of the playback.

This guide shows a couple of great ways to hack your Walkman for speed control. First there's the Walkmellotron, which uses USB MIDI on a QT Py RP2040 with a DAC board for voltage control of the player's potentiometer wiper. Use a MIDI keyboard, controller, or sequencer to "play" tape tunes.
The second method replaces the tape player’s on-board motor controller entirely, allowing total motor control, including a wide range of speeds in both forward and reverse! NOTE: custom tape loop cassette required.

Walkmellotron MIDI Controlled Tape Player

The Walkmellotron uses a widely available GE cassette player that featured a simple speed control dial. To adjust the playback speed, you'll connect a DAC (digital-to-analog converter) to the ground and wiper of the speed dial potentiometer, and send it different voltages to select the speed. The QT Py RP2040 will be used to interpret USB MIDI messages and send the voltages via the STEMMA QT-connected DAC board.

This build is loosely inspired by the Mellotron, an amazing instrument invented in 1963, that used one tape loop per key to play beautiful, lush soundscapes.
Parts

**Adafruit QT Py RP2040**
What a cutie pie! Or is it... a QT Py? This diminutive dev board comes with one of our new favorite chip, the RP2040. It's been made famous in the new [https://www.adafruit.com/product/4900](https://www.adafruit.com/product/4900)

**Adafruit MCP4728 Quad DAC with EEPROM**
If you've ever said to yourself, "Gee, I wish I had four 12-bit DACs that came in a single package with the ability to save their settings to an EEPROM", well I have good... [https://www.adafruit.com/product/4470](https://www.adafruit.com/product/4470)

**Pink and Purple Woven USB A to USB C Cable - 1 meter long**
This cable is not only super-fashionable, with a woven pink and purple Blinka-like pattern, it's also made for USB C for our modernized breakout boards, Feathers, and... [https://www.adafruit.com/product/5153](https://www.adafruit.com/product/5153)
STEMMA QT / Qwiic JST SH 4-Pin Cable - 50mm Long
This 4-wire cable is 50mm / 1.9" long and fitted with JST SH female 4-pin connectors on both ends. Compared with the chunkier JST PH these are 1mm pitch instead of 2mm, but...
https://www.adafruit.com/product/4399

Cassette Player

GE 3-5362A Cassette Player/Recorder
This is a great, easily available player to use for cassette hacks. Online auction sites are full of them, usually priced to sell at around $10-$20.

Look for one that has been tested and works, unless you want to get a replacement belt, which is the most common thing to break in old tape players.
Build the Walkmellotron

The player needs to be modified to receive control voltage from the DAC board. This is as simple as soldering a wire to cassette player ground electrical (GND) and another to the variable speed control potentiometer's wiper, which is labeled VS+ (variable speed positive, perhaps) on the board.

To start, remove the four screws on the back of the player and carefully open it up, being sure not to damage the battery and speaker wires.
Solder to VS+
Solder a 6" length of red silicone covered wire to the VS+ point. This will go to the VA (voltage A) pin of the DAC board.
Solder to GND

Next, solder a 6" length of blue wire to a GND point on the player as shown here. (You can find a GND point with a multimeter in continuity mode testing from a known ground if needed, such as the negative battery contact.)

This wire will go to the GND pad on the DAC.

Wire Run

Run the two wires you soldered outside of the player via the nearby DC jack hole.

NOTE: If you plan to use external power, you can run them through a gap in the cassette well, or drill a separate hole for the wires.

Use some short lengths of heat shrink tubing to neaten up the wiring if you like.
Wire to DAC
Solder the wires to the DAC board as shown:
blue wire to DAC GND pin
red wire to DAC VA pin

DAC to QT Py, come in QT Py
Use the STEMMA QT cable to connect the DAC board to the QT Py RP2040.
Affix!
Use two adhesive squares to affix the DAC and the QT Py to the case as shown.
Plug in the USB C cable to your computer and the QT Py so you can get ready to code it!

CircuitPython

CircuitPython () is a derivative of MicroPython () designed to simplify experimentation and education on low-cost microcontrollers. It makes it easier than ever to get
prototyping by requiring no upfront desktop software downloads. Simply copy and edit files on the CIRCUITPY drive to iterate.

CircuitPython Quickstart

Follow this step-by-step to quickly get CircuitPython running on your board.

Download the latest version of CircuitPython for this board via circuitpython.org

Click the link above to download the latest CircuitPython UF2 file.

Save it wherever is convenient for you.

To enter the bootloader, hold down the BOOT/BOOTSEL button (highlighted in red above), and while continuing to hold it (don't let go!), press and release the reset button (highlighted in blue above). Continue to hold the BOOT/BOOTSEL button until the RPI-RP2 drive appears!
If the drive does not appear, release all the buttons, and then repeat the process above.

You can also start with your board unplugged from USB, press and hold the BOOTSEL button (highlighted in red above), continue to hold it while plugging it into USB, and wait for the drive to appear before releasing the button.

A lot of people end up using charge-only USB cables and it is very frustrating! Make sure you have a USB cable you know is good for data sync.

You will see a new disk drive appear called RPI-RP2.

Drag the adafruit_circuitpython_etc.uf2 file to RPI-RP2.

The RPI-RP2 drive will disappear and a new disk drive called CIRCUITPY will appear.

That's it, you're done! :)

©Adafruit Industries
Safe Mode

You want to edit your code.py or modify the files on your CIRCUITPY drive, but find that you can't. Perhaps your board has gotten into a state where CIRCUITPY is read-only. You may have turned off the CIRCUITPY drive altogether. Whatever the reason, safe mode can help.

Safe mode in CircuitPython does not run any user code on startup, and disables auto-reload. This means a few things. First, safe mode bypasses any code in boot.py (where you can set CIRCUITPY read-only or turn it off completely). Second, it does not run the code in code.py. And finally, it does not automatically soft-reload when data is written to the CIRCUITPY drive.

Therefore, whatever you may have done to put your board in a non-interactive state, safe mode gives you the opportunity to correct it without losing all of the data on the CIRCUITPY drive.

Entering Safe Mode

To enter safe mode when using CircuitPython, plug in your board or hit reset (highlighted in red above). Immediately after the board starts up or resets, it waits 1000ms. On some boards, the onboard status LED (highlighted in green above) will blink yellow during that time. If you press reset during that 1000ms, the board will start up in safe mode. It can be difficult to react to the yellow LED, so you may want to think of it simply as a slow double click of the reset button. (Remember, a fast double click of reset enters the bootloader.)

In Safe Mode

If you successfully enter safe mode on CircuitPython, the LED will intermittently blink yellow three times.

If you connect to the serial console, you'll find the following message.

```
Auto-reload is off.
Running in safe mode! Not running saved code.
CircuitPython is in safe mode because you pressed the reset button during boot.
Press again to exit safe mode.
Press any key to enter the REPL. Use CTRL-D to reload.
```
You can now edit the contents of the CIRCUITPY drive. Remember, your code will not run until you press the reset button, or unplug and plug in your board, to get out of safe mode.

Flash Resetting UF2

If your board ever gets into a really weird state and doesn't even show up as a disk drive when installing CircuitPython, try loading this 'nuke' UF2 which will do a 'deep clean' on your Flash Memory. You will lose all the files on the board, but at least you'll be able to revive it! After loading this UF2, follow the steps above to re-install CircuitPython.

[Download flash erasing "nuke" UF2]

Code the Walkmellotron

First make sure you are running the latest version of Adafruit CircuitPython for your board.

Text Editor

Adafruit recommends using the Mu editor for using your CircuitPython code with the Feather. You can get more info in this guide.

Alternatively, you can use any text editor that saves text files.

Download the Project Bundle

Your project will use a specific set of CircuitPython libraries and the code.py file. To get everything you need, click on the Download Project Bundle link below, and uncompressed the .zip file.

Drag the contents of the uncompressed bundle directory onto your QT Py board's CIRCUITPY drive, replacing any existing files or directories with the same names, and adding any new ones that are necessary.
Next you can hook up your Walkmellotron and play!

Jam on the Walkmellotron

Source

A continuous tone or chord is great source material for your Walkmellotron tapes. Record an entire side of a cassette with this tone or chord.

For this demo, I recorded a slightly de-tuned sine wave tritone I made with VCV Rack.
Playing the Walkmellotron
Place your prepared tape into the Walkmellotron and make sure it is rewound.

Hook up the audio out to headphones or an amplifier/powered speakers -- or even better through some effects pedals or outboard units such as reverb and delay.

Plug in the QT Py to a USB MIDI host/controller/sequencer or your computer/iOS device. Basically, anything capable of being a USB MIDI host.

I used a RetroKits RK-006 to pair the Walkmellotron with an Arturia Keystep MIDI keyboard.
Press play to start the cassette playing.

Now, play any notes from MIDI 1 to 31 (A0 - G1). You can adjust the note range in code.py if needed, depending on the range of your MIDI controller.

Your Walkmellotron will play its mellow/slightly haunting tones!
Motor Controlled Cassette Player

This cassette player modification controls the player's motor directly with an L9110 motor driver. An Adafruit QT Py RP2040 controls the L9110 with two PWM pins and uses CircuitPython code to affect the motor's throttle by reading a potentiometer value.

This mod should be used with a cassette tape loop. By driving the motor independently from the cassette player's circuit board, you can damage regular cassette tapes since the motor will keep spinning the tape even once the tape has reached the end.

Make sure to use this modification with a cassette tape loop! Otherwise you may damage regular cassette tapes.
The circuit is built on a perma-proto board that fits perfectly on the back of the cassette player with a 3D printed plate. The QT Py RP2040 can power the audio output of the player with its 3.3V output.

The potentiometer affects the speed and direction of the motor being driven by the L9110 driver. If you turn the potentiometer all the way to the left, the motor throttles at full speed in reverse. If you turn the potentiometer all the way to the right, the motor throttles forward at full speed.

The button controls a revved warble effect when it is held down. The motor will perform the effect forward or reversed depending on the position of the potentiometer.
Adafruit QT Py RP2040
What a cutie pie! Or is it... a QT Py? This diminutive dev board comes with one of our new favorite chip, the RP2040. It's been made famous in the new
https://www.adafruit.com/product/4900

L9110H H-Bridge Motor Driver for DC Motors - 8 DIP
Run two solenoids or a single DC motor with up to 800mA per channel using the super-simple L9110H H-bridge driver. This bridge chip is an 8 DIP package so it's easy to fit onto any...
https://www.adafruit.com/product/4489

Adafruit Perma-Proto Half-sized Breadboard PCB - Single
Customers have asked us to carry basic perf-board, but we never liked the look of most basic perf: it's always crummy quality, with pads that flake off and no labeling. Then we...
https://www.adafruit.com/product/1609
Alpha Single-Gang 16mm Right-angle PC Mount - 10K Linear
This Alpha 10K Linear potentiometer is a two-in-one, good in a breadboard or with a panel. It's a fairly standard linear taper 10K ohm potentiometer. It's smooth and... https://www.adafruit.com/product/5280

Soft Silicone Top 6mm Push-buttons (20-pack)
This product is really annoying to have to write a description for because it's really something you have to touch to understand! Instead of a hard plastic actuator-button, these... https://www.adafruit.com/product/4183

Breadboard-friendly SPDT Slide Switch
These nice switches are perfect for use with breadboard and perfboard projects. They have 0.1" spacing and snap in nicely into a solderless breadboard. They’re easy to switch... https://www.adafruit.com/product/805

| 1 x IC Socket - for 8-pin 0.3" Chips - Pack of 3 | https://www.adafruit.com/product/2202 |
| DIP-8 IC socket |
| 2 x 2.54mm/0.1" Pitch Terminal Block - 2-pin | https://www.adafruit.com/product/2138 |
| 2-pin terminal block |
| 1 x USB-C Cable | https://www.adafruit.com/product/5153 |
| USB-C to USB-A cable |
Cassette Gear

1 x **Portable Cassette Player**
Portable cassette player


1 x **Maxell UR 90 Normal Bias Blank Audio Recording Cassette Tape**
Blank cassette tapes

[https://www.amazon.com/gp/product/B00006IAAL/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1](https://www.amazon.com/gp/product/B00006IAAL/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1)

1 x **Audio Cassette Tape Splicing Tape**
Cassette splicing tape

[https://www.amazon.com/gp/product/B00GK77D1G/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1](https://www.amazon.com/gp/product/B00GK77D1G/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1)

### Circuit Diagram

![Circuit Diagram](fritzing)

**Button**

- Board pin A0 to button input
- Board GND to button GND

**Potentiometer**

- Board GND to potentiometer pin 1
- Board pin A1 to potentiometer wiper

© Adafruit Industries
Board 3.3V to potentiometer output

Switch

- Board 5V to switch input

L9110 IC

- Cassette motor wire 1 to L9110 pin 1
- Switch output to L9110 pin 2
- Switch output to L9110 pin 3
- Cassette motor wire 2 to L9110 pin 4
- Board GND to L9110 pin 5
- Board pin A3 to L9110 pin 6
- Board pin A2 to L9110 pin 7
- Board GND to L9110 pin 8

Cassette Power Terminal Block

- Board 3.3V to cassette player positive terminal
- Board GND to cassette player negative terminal

3D Printing

The perma-proto board may be mounted on a 3D printed plate, described below. The plate benefits from some supports. There is a 3D printed lever for the potentiometer that prints with no supports.
The STL files can be downloaded directly here or from Thingiverse.

You can use support blockers in your slicer software to only have supports in the center of the plate's mount. This should provide enough support for your printer to bridge the wall.

The plate slides onto the cassette player's belt clip to keep the mod compact.

The potentiometer lever is sized to fit onto the circular alpha pot.
Code the Motor Controlled Cassette Player

Once you've finished setting up your QT Py RP2040 with CircuitPython, you can access the code and necessary libraries by downloading the Project Bundle.

To do this, click on the Download Project Bundle button in the window below. It will download as a zipped folder.

```python
import time
import board
import pwmio
import simpleio
from adafruit_motor import motor
from analogio import AnalogIn
from digitalio import DigitalInOut, Direction, Pull

# button setup
warble_switch = DigitalInOut(board.A0)
warble_switch.direction = Direction.INPUT
warble_switch.pull = Pull.UP

# potentiometer setup
pot = AnalogIn(board.A1)

# PWM pins for L9110
PWM_PIN_A = board.A3  # pick any pwm pins on their own channels
PWM_PIN_B = board.A2

# PWM setup
pwm_a = pwmio.PWMOut(PWM_PIN_A, frequency=50)
pwm_b = pwmio.PWMOut(PWM_PIN_B, frequency=50)

# motor setup
cassette = motor.DCMotor(pwm_a, pwm_b)

# variables for warble switch
i = 0.4
last_i = 0.4
pos = False
neg = False

while True:
    # map range of pot to range of motor speed
    # all the way to the left will run the motor in reverse full speed
    # all the way to the right will run the motor forward full speed
    mapped_speed = simpleio.map_range(pot.value, 0, 65535, -1.0, 1.0)
    # if you press the button...
    if not warble_switch.value:
        # checks current pot reading
        # if it's positive...
        if mapped_speed > 0:
            # sets starting speed
            i = 0.4
            # sets last value to loop
            last_i = i
            # notes that it's positive
            pos = True
            # if it's negative...
        else:
            # sets starting speed
            i = -0.4
```
# sets last value to loop
last_i = i
# notes that it's negative
neg = True
# loop 8 times
for z in range(8):
    # if it's positive...
    if pos:
        # increase speed
        i += 0.06
    # if it's negative
    else:
        # decrease speed
        i -= 0.06
    # send value to motor
    cassette.throttle = i
    time.sleep(0.1)
# loop the value while button is held down
i = last_i
pos = False
neg = False
# run motor at mapped speed from the pot
 cassette.throttle = mapped_speed

Upload the Code and Libraries to the QT Py RP2040

After downloading the Project Bundle, plug your QT Py RP2040 into the computer's USB port with a known good USB data+power cable. You should see a new flash drive appear in the computer's File Explorer or Finder (depending on your operating system) called CIRCUITPY. Unzip the folder and copy the following items to the QT Py RP2040's CIRCUITPY drive.

- lib folder
- code.py

Your QT Py RP2040 CIRCUITPY drive should look like this after copying the lib folder and the code.py file.
How the CircuitPython Code Works

The L9110 motor controller can be used with the `adafruit_motor` CircuitPython library. It just requires two PWM pins.

```python
# PWM pins for L9110
PWM_PIN_A = board.A3  # pick any pwm pins on their own channels
PWM_PIN_B = board.A2
# PWM setup
pwm_a = pwmio.PWMOut(PWM_PIN_A, frequency=50)
pwm_b = pwmio.PWMOutput(PWM_PIN_B, frequency=50)
# motor setup
 cassette = motor.DCMotor(pwm_a, pwm_b)
```

In the loop, the potentiometer's value range is mapped to the motor's throttle range. When the pot's value is 0, the motor will throttle full speed in reverse. When the pot's value is 65535, the motor will throttle full speed forward.

```python
mapped_speed = simpleio.map_range(pot.value, 0, 65535, -1.0, 1.0)
cassette.throttle = mapped_speed
```

The button acts as a revved warble switch. When it's pressed, it incrementally increases the motor's throttle in increments of 0.06 to create a warbling sound. The direction of the warble is affected by the pot's position.

```python
# if you press the button...
 # creates a ramping effect
 if not warble_switch.value:
    # checks current pot reading
    # if it's positive...
    if mapped_speed > 0:
        # sets starting speed
        i = 0.4
        # sets last value to loop
        last_i = i
        # notes that it's positive
        pos = True
        # if it's negative...
    else:
        # sets starting speed
        i = -0.4
        # sets last value to loop
        last_i = i
        # notes that it's negative
        neg = True
    # loop 8 times
    for z in range(8):
        # if it's positive...
        if pos:
            # increase speed
            i += 0.06
        # if it's negative
        else:
            # decrease speed
            i -= 0.06
        # send value to motor
```
```python
cassette.throttle = i
time.sleep(0.1)
# loop the value while button is held down
i = last i
pos = False
neg = False
```

## Soldering

![Soldered circuit](image)

The circuit is soldered to a half size perma-proto board. You can reference the perma-proto board’s coordinate numbers to make sure all of the items will fit.

## Components

### Socket Headers

Solder socket headers to the perma-proto board for the QT Py RP2040. You can use the QT Py as a jig for the headers. The headers will be placed at perma-proto board coordinates D1-D7 and H1-H7.
DIP-8 Socket
Solder a DIP-8 socket to the perma-proto board with pin 1 placed at coordinate E17.

Terminal Blocks
Solder the motor's terminal block at coordinates A14 and A15. Solder the cassette player's power terminal block on the front power and ground rails.

Button
Solder the button with pin 1 at coordinate E26.
Switch
Solder the switch with pin 1 at coordinate J23.

Potentiometer
Solder the potentiometer with the ground pin at coordinate J10, wiper at J12 and pin 3 at J14.

Wiring
GND Signals
Solder the following ground connections with solid core wire.

GND rail to I2 (board GND)
GND rail to I10 (potentiometer GND)
GND rail to I17 (L9110 pin 8)
GND rail to I20 (L9110 pin 5)
GND rail to I28 (button GND)
GND rail to GND rail
3.3V Signals
Solder the following 3.3V connections with solid core wire.

Front positive rail to I3 (board 3.3V)
Front positive rail to I14 (potentiometer pin 3)

5V Signals
Solder the following 5V connections with solid core wire.

Back positive rail to I1 (board 5V)
Back positive rail to H25 (switch input)
H24 to B19 (switch output to L9110 pin 3)
B19 to B18 (L9110 pin 3 to L9110 pin 2)

Button Input
Solder a piece of wire from A26 (button input) to A1 (board pin A0).
Potentiometer Wiper
Solder a piece of wire from H12 (potentiometer wiper) to A2 (board pin A1).

PWM Connections
Solder the following PWM connections for the L9110.
H18 (L9110 pin 7) to C3 (board pin A2)
G19 (L9110 pin 6) to B4 (board pin A3)

Motor Connections
Solder the following motor connections for the L9110.
C20 (L9110 pin 4) to C15 (motor terminal block)
D17 (L9110 pin 1) to D14 (motor terminal block)
Cassette Player Wiring

Open the cassette player by removing the four screws on the back.

Motor Wiring

Desolder the cassette player motor's two wires from the PCB.

Extend the motor wires' lengths with pieces of wire and heatshrink over the solder connections.
Power Connections

Cut and splice two pieces of wire for the cassette player's battery terminals.

Solder one wire to the back of the positive battery terminal.

Solder one wire to the back of the negative battery terminal.
Now you'll have four wires connected to the cassette player: two for the motor and two for the battery terminals.

Assembly

Run the four breakout wires through the cassette player's battery door. Reattach the back of the player with the four screws.

Attach the perma-proto board circuit to the 3D printed plate with M2.5 screws and standoffs.
Slide the plate's notch over the cassette player's belt clip.

Attach the motor's wires to the motor terminal block. Attach the player's power wires to the power terminal block. Make sure that the positive wire is attached to the positive terminal and that the negative wire is attached to the negative terminal.

Attach the potentiometer lever and you're ready to start playing with tape loops!
Make a Cassette Tape Loop

There are cassette tape loops available for purchase but it is fairly simple to create your own.

Open the blank cassette's case by removing the four screws.
Remove the two large wheels in the cassette body. Push the roll of magnetic tape off of the wheel. Then, cut the magnetic tape from the wheels using an exacto knife.

Be careful when cutting the magnetic tape!

Cut a piece of magnetic tape that is approximately 8 5/8 inches long.
Join the two ends of the tape together using cassette splicing tape. Place the tape on the inside of the loop.

Run the magnetic tape loop through the bottom of the cassette tape housing. Finish installing the loop by placing the loop over the wheel on the right.

Replace the wheel on the left inside the cassette housing. Close the housing with the four screws. Now you're ready to record a loop!
Usage

Make sure to use this modification with a cassette tape loop! Otherwise you may damage regular cassette tapes.

To use the modified cassette player, first you will press play on the player and then switch on the L9110 circuit with the slide switch.

You can use the built-in speaker or a standard 1/8" audio jack cable to get audio out from your tape loop. If you use the headphone jack, you can send the audio signal through effects pedals like reverb and delay.
Turn the potentiometer to affect playback speed and direction. Press the button to get the warbling speed up effect.

The motor speed can also be used while recording to a tape loop. This can give you more or less recording time, as well as unique effects that you wouldn't be able to normally record.