Minibot

Release 0.1

Dec 30, 2020

Setup RPI and Crickit

- Minibot GitHub : https://github.com/Atikul99/minibot

CHAPTER 1

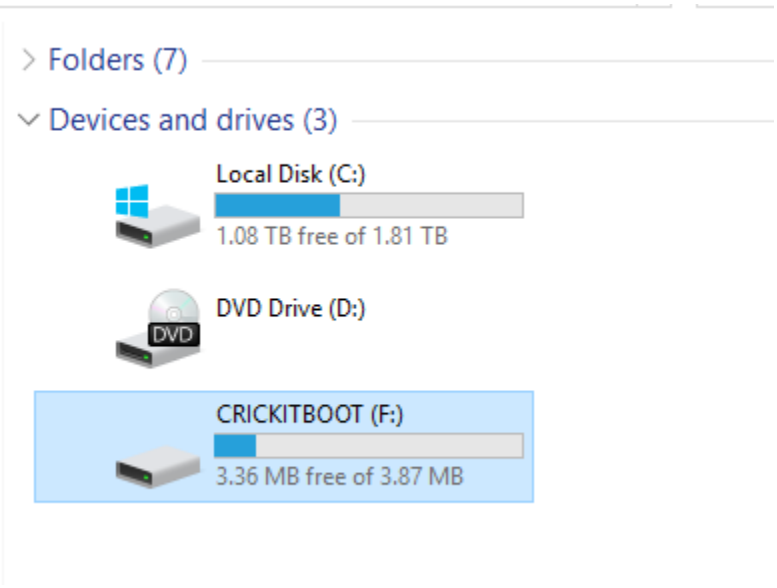
Crickit Prerequisites

Setup Crickit for development

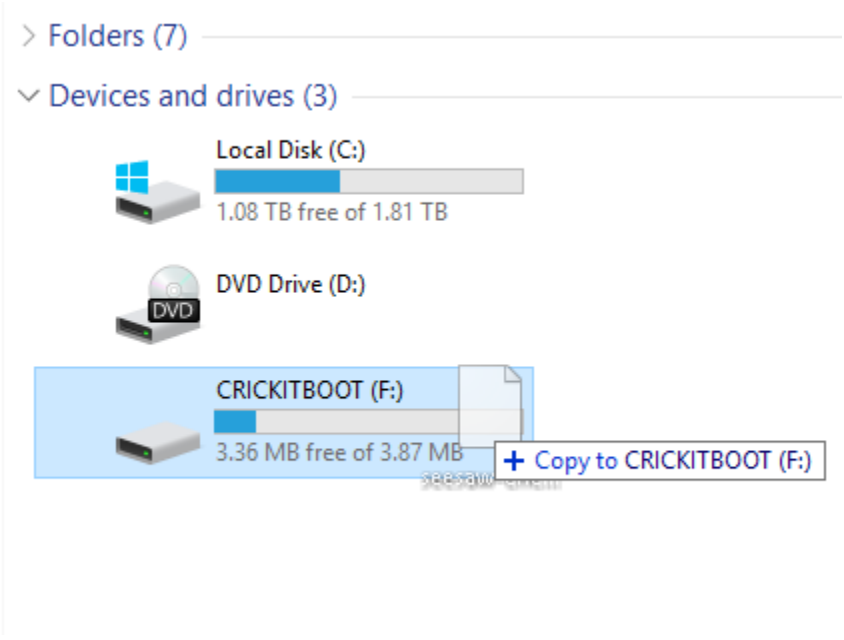
Note: No additional power supply is needed for the following steps

1.1 Update the Crickit

1. Plug in USB cable into seesaw/Crickit
2. Double-click the Crickit Reset button
3. Look for pulsing yellow LED
4. Look for a New Disk on Your Computer [CRICKITBOOT]



5. Download the latest firmware
6. Drag UF2 file onto CRICKITBOOT



7. Party!

1.2 Power Supply

Now plug in the included battery bank for power supply

Hint: *For more help, refer to* adafruit's *website for further guidance*

CHAPTER 2

Raspberry-Pi Prerequisites

Please run the following commands to setup the raspberry pi for code deployment

Note: Connect to a display using HDMI to run the code

Raspberry Pi password = `raspberry`

2.1 Power Supply

Use the included battery bank for power supply

2.2 Update distribution packages

First lets update your distribution packages

```
sudo apt-get update;
sudo apt-get upgrade;
sudo reboot;
```

2.3 Install git, python3 & pip

Next run the following commands to install git, python3 & pip

Attention: *Python3 is required*

```
sudo apt-get install -y python3 git python3-pip
```

2.4 Install CRICKIT Python3 lib.

Next run the following commands to install CRICKIT Python 3 libraries

```
sudo pip3 install RPI.GPIO adafruit-blinka
```

2.5 Install smbus i2c-tools

Next run the following commands to add SMBus and I2C support to Python

```
sudo apt-get install python-smbus;
sudo apt-get install -y i2c-tools
```

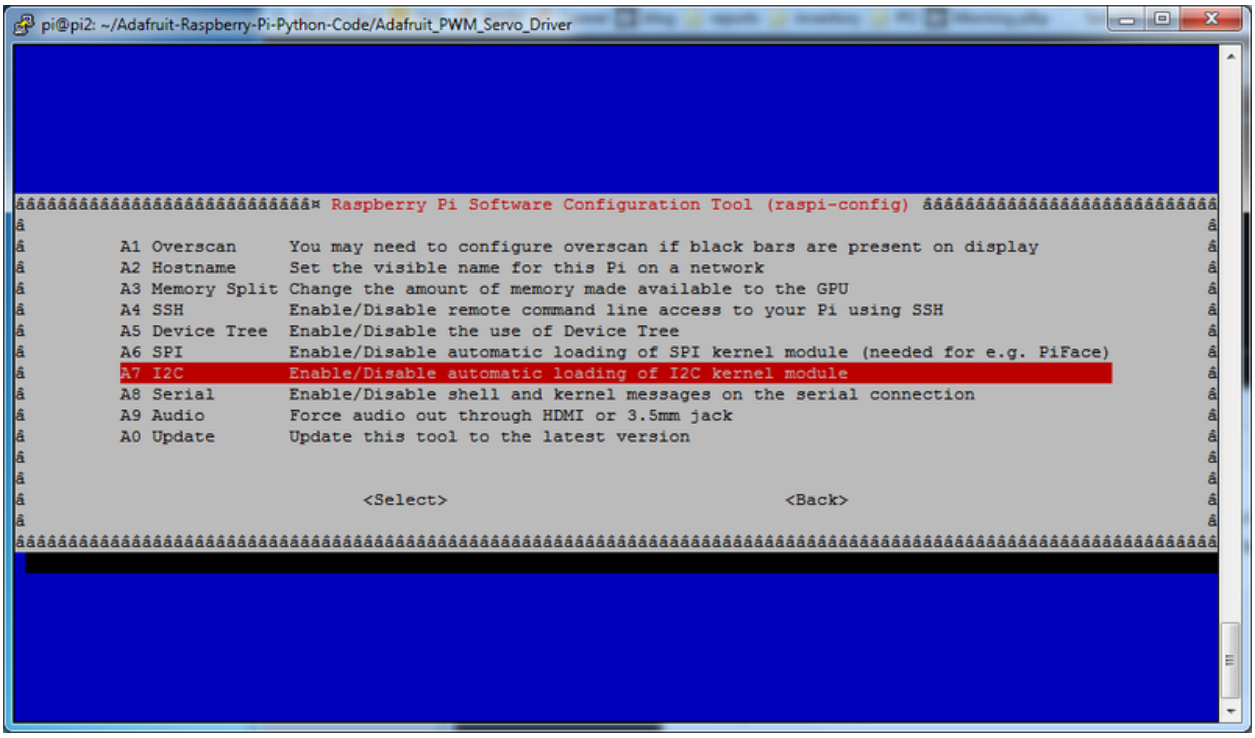
2.6 I2C Kernal Support

Installing Kernel Support for I2C devices

Run the following command for gui interface

```
sudo raspi-config
```

On the GUI select **Interfacing Options** followed by **I2C**



When prompted to Enable I2C select ''yes''

Reboot device to ensure I2C device support

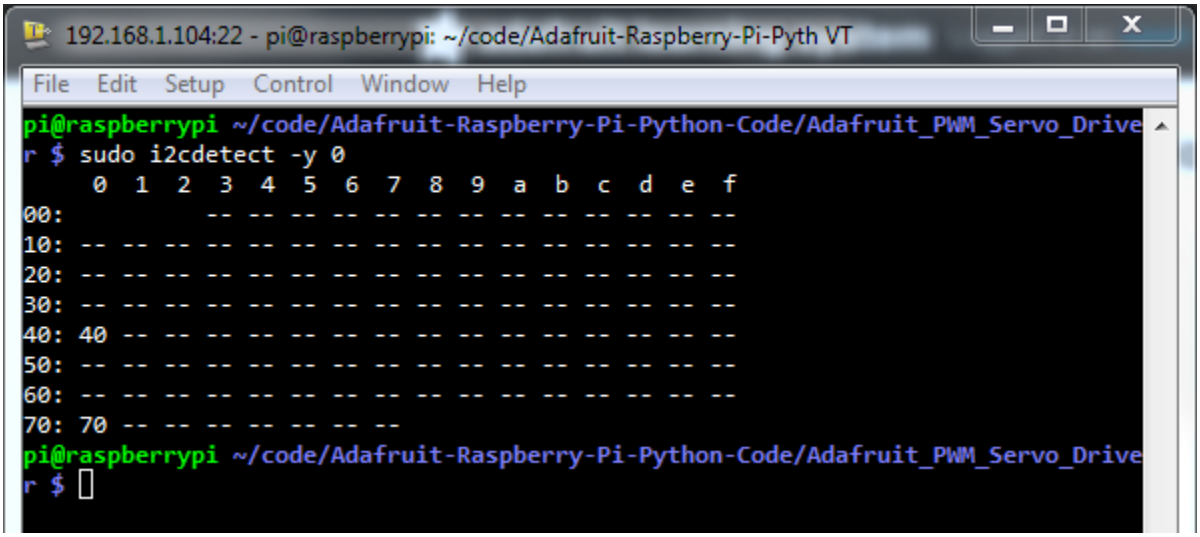
To reboot run the following command

```
sudo reboot
```

Upon boot run the following command to see all the connected devices

```
sudo i2cdetect -y 1
```

it should show up at 0x40 (binary 1000000) as follows:



2.7 Install adafruit-circuitpython-crickit

To interact with the servos, install adafruit-circuitpython-crickit by running the following command

```
sudo pip3 install adafruit-circuitpython-crickit
```

2.8 Run all at once

```
# Installations
sudo apt-get update;
sudo apt-get upgrade;
sudo reboot;
sudo apt-get install -y python3 git python3-pip;
sudo pip3 install RPI.GPIO adafruit-blinka;
sudo apt-get install python-smbus;
sudo apt-get install -y i2c-tools;
# User input is required [select Interfacing Options>>I2C>>yes]
sudo raspi-config;

# Need to run separately
sudo reboot;
sudo i2cdetect -y 1
```

Hint: *For more help, refer to* adafruit's *website for further guidance*

CHAPTER 3

Motors and Servos

Ports for plugging in Motors and Servos

3.1 DC-Motors

Note: Crickit HAT only supports 2 DC Motors

Use with Motor port on the Crickit HAT

3.2 Servos and Continious Servos

Note: Crickit HAT only supports 4 DC Motors

Use with Servo port on the Crickit HAT

CHAPTER 4

Setup

In order to start coding download the package minilib

4.1 Install minilib

Lets install minilib

```
pip3 install minilib
```

4.2 Party!

You're done

CHAPTER 5

Sample code

Sample code utilizing minilib package

`Test driving`

```
from minilib import Joystick, ArcadeDrive, Motor

left = Motor(0)
right = Motor(1)
jstick = Joystick(0)
robot = ArcadeDrive(left, right)

while True:
    forwardAxis = jstick.getAxis(1)
    steerAxis = jstick.getAxis(3)
    robot.drive(forwardAxis, steerAxis)
```

`Test Motors`

```
import time
from minilib import Motor

motor_1 = Motor(0)
motor_2 = Motor(1)

for _ in range(3):
    motor_1.throttle(1)
    motor_2.throttle(-1)
    time.sleep(1)

    motor_1.throttle(-0.5)
    motor_2.throttle(0.5)
    time.sleep(1.5)
```

(continues on next page)

(continued from previous page)

```
    motor_1.throttle(1)
    motor_2.throttle(-1)
    time.sleep(1)
```

Test Servos

```
import time
from minilib import Servo

servo_1 = Servo(0)
servo_2 = Servo(1)

for _ in range(3):
    servo_1.angle(180)
    servo_2.angle(-180)
    time.sleep(1)

    servo_1.angle(-90)
    servo_2.angle(90)
    time.sleep(1.5)

    servo_1.angle(180)
    servo_2.angle(-180)
    time.sleep(1)
```

CHAPTER 6

minilib Library

Documentation for the Library

6.1 minilib.ArcadeDrive

class minilib.Drive.**ArcadeDrive**(*self*, *left*, *right*)

__init__(*self*, *left*, *right*)
Setup Arcade drive

Parameters

- **left** – left side drive
- **rightServo** (*continuous_servo*) – right side drive

drive(*self*, *forwardPower*, *steerPower*)
Drive given the forward and steer axis power Meant to be run inside a while loop

Parameters

- **forwardPower** (*float*) – Forward power from joystick axis
- **steerPower** (*float*) – Steer power from joystick axis

6.2 minilib.TankDrive

class minilib.Drive.**TankDrive**(*self*, *left*, *right*)

__init__(*self*, *left*, *right*)
Setup Arcade drive

Parameters

- **left** – left side drive
- **rightServo** (*continuous_servo*) – right side drive

drive(*self*, *leftPower*, *rightPower*)

Drive given the left and right axis power Meant to be run inside a while loop

Parameters

- **leftPower** (*float*) – Forward power from joystick axis
- **rightPower** (*float*) – Steer power from joystick axis

6.3 minilib.Joystick

class minilib.Joystick.**Joystick**(*self*, *ID*, *deadband=0*)

__init__(*self*, *ID*, *deadband=0*)
Setup Joystick control using pygame :param id: ID of the joystick :type id: int

getAxis(*self*, *axisID*)
Get the value of the axis

Parameters **axisID** (*int*) – Id of the axis

Returns value of the axis

Return type float

getButton(*self*, *buttonID*)
Get the state of the button

Parameters **buttonID** (*int*) – Id of the button

Returns state of the button

Return type bool

6.4 minilib.Motor

class minilib.Motor.**Motor**(*self*, *ID*)

__init__(*self*, *ID*)
Inintialize the DC Motor

Parameters **ID** (*int*) – The ID of the Motor [0,1]

throttle(*self*, *power*)
Input power for the Motor

Parameters **power** (*float*) – Value from -1 to 1

6.5 minilib.Servo

class minilib.Servo.**Servo**(*self*, *ID*)

__init__(*self*, *ID*)
Initialize a Servo

Parameters **ID** (*int*) – ID of the Servo [0,1,2,3]

angle(*self*, *degree*)
Set the angle to rotate to

Parameters **degree** (*int*) – degree of the Servo

6.6 minilib.Servo.ContiniousServo

class minilib.Servo.**ContiniousServo**(*self*, *ID*)

__init__(*self*, *ID*)
Initialize a Continious Servo

Parameters **ID** (*int*) – ID of the Continious Servo [0,1,2,3]

throttle(*self*, *power*)
Set the throttle of the Continious Servo

Parameters **power** (*float*) – Power of the Continious Servo -1 to 1

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

Index

Symbols

A

D

G

M

T