Building a self watering plant using Micropython on a WIFI enabled arduino esp8266

Anele Makhaba
Junior Software Engineer
How it started
Why

Young & Curious

I'm a Maker.

Mpho Mphego

IoT, Raspberry Pi, Arduino, MicroPython, ESP8266/ESP32.
Overview

- What is Micropython/uPython
- Micropython vs Circuit Python
- Arduino vs Micropython
- Micropython in Microcontrollers
- Demo *(Self Watering Plant)*
  - How
  - Information Insights.
What is Micropython/uPython

- Tiny open source python programming language interpreter
- Runs on small embedded development boards like the bbc micro bit

Capabilities
- Control hardware & connected devices
- Read analogue sensors
- Network & wifi Support (Board dependant)
- And many more

Features
- Interactive REPL (Read Eval Print Loop)
- Extensive Software library
- Extensibility
Micropython and Circuit Python

- Circuit python is adafruit's open source derivative of micropython
- Adds support for easily getting started with electronics
- Core language is the same for micropython and circuit python
- Slightly differ in api`s for accessing hardware components
Arduino vs MicroPython

- Entire Ecosystem
- Compiled programming language
- Maximum performance saving memory

- Just a programming language
- Interpreted Programming language
- Less performance and more memory usage

With Micropython you can write clean and simple python code instead of having to use complex low level languages like C (what arduino uses).
Micropython in Microcontrollers

- Flash firmware
- Using micropython in REPL (Read Eval Print Loop) prompt
- Using micropython in WEBREPL - Prompt via WIFI (Microcontrollers with WIFI)
- Using a text editor extensions - VSCODE
- Main scripts
- Manipulating Files & Running micropython code
Flash firmware

- Download latest Micropython Firmware
- Put microcontroller in boot-loader mode (Process is board specific.)
- boot-loader mode will allow a swift flashing of the new firmware
- esptool is a utility used to communicate with ROM bootloader in esp8266 & esp32 series chips
  - Available via “pip” or clone the github repo.
- Recommended to erase entire flash of microcontroller before new firmware update.
  - esptool example
    - esptool.py --port <tty port device> erase_flash
Micropython in Microcontrollers
Flash firmware Cont...

☐ Flash downloaded firmware using the esptool(linux) or similar alternative for other OS.

☐ esptool example

```bash
esptool.py --port <tty port device> --baud <baudrate> write_flash --flash_size=detect 0 <new-firmware>.bin
```
Micropython in Microcontrollers

REPL (Read Eval Print Loop) prompt

☐ Using micropython in REPL prompt
   use terminal emulator program like Teraterm (windows), picocom (linux) similar in other operating systems to access REPL
   picocom <tty port device> -b<baudrate>

   ```
   picocom /dev/ttyUSB0 -b115200
   ```

☐ Using micropython in WEBREPL - Prompt via WIFI
   ☐ WebREPL client is hosted [http://micropython.org/webrepl](http://micropython.org/webrepl)
   ☐ configure microcontroller as an AP before connecting to WEBREPL
   ☐ To access WEBREPL connect computer to microcontroller`s AP.
Micropython in Microcontrollers

Using a text editor extensions - VSCODE

☐ Download Micropython IDE extension
  ☐ Launch VS Code Quick Open and paste the command
    `ext install dphans.micropython-ide-vscode`

☐ Requirements
  ☐ ampy - utility that allows you to interact with file system created on the chip
  ☐ rshell - Remote shell for micropython

☐ Features
  ☐ Manage projects
  ☐ Run and Stop scripts
  ☐ Flash new firmware
Micropython in Microcontrollers

Main scripts

*boot.py*

- Script that runs when board boot up or wake up from sleep
- Should contain low level code that sets up the board

*main.py*

- If exist, it runs *boot.py*
- Should contain any main script that needs to run when the board is powered up or reset
Manipulating Files & Running micropython code

- Ampy is a utility that allows you to interact with file system created on the chip.
- We`ll use ampy to show how to:
  - Manipulate Files
  - Run micropython code

- Ampy is a simple cross-platform command line tool that offers enough functionality without being too complex
- There are alternatives like rshell for manipulating files and more on a microcontroller.
Micropython in Microcontrollers

Usage: ampy [OPTIONS] COMMAND [ARGS]...

ampy - Adafruit MicroPython Tool

Ampy is a tool to control MicroPython boards over a serial connection. Using ampy you can manipulate files on the board's internal filesystem and even run scripts.

Options:
- `p`, `--port PORT` Name of serial port for connected board. Can optionally specify with AMPY_PORT environment variable. [required]
- `b`, `--baud BAUD` Baud rate for the serial connection (default 115200). Can optionally specify with AMPY_BAUD environment variable.
- `d`, `--delay DELAY` Delay in seconds before entering RAW MODE (default 0). Can optionally specify with AMPY_DELAY environment variable.
- `--version` Show the version and exit.
- `--help` Show this message and exit.

Commands:
- `get` Retrieve a file from the board.
- `ls` List contents of a directory on the board.
- `mkdir` Create a directory on the board.
- `put` Put a file or folder and its contents on the board.
- `reset` Perform soft reset/reboot of the board.
- `rm` Remove a file from the board.
- `rmdir` Forcefully remove a folder and all its children from the board.
- `run` Run a script and print its output.
Demo (Self Watering Plant)

What you need

- ESP8266 (NodeMCU) Board
- Moisture Sensor
- Micro USB to USB cable
- Water Pump
- Dopunt F-F(jump) wires
- Connectors for pump
- Transistor TIP31C
- Plant and pot for the plant
How

- Setup esp8266 board (Flash firmware)
- Watering system diagram
- Calibrate moisture sensor
- Setup water pump
- Group everything together.
How

Watering system diagram
How

Calibrate moisture sensor

☐ Connect the sensor to the board.
   ☐ Check the above diagram see where it will connect to the board
   ☐ Dip the sensor in a bowl of water and take the reading.
   ☐ Wipe the sensor completely dry and take the reading.

☐ The sensor has both an analogue (0 for wet and 1024 for completely dry) and a digital (LOW / 0 for wet and HIGH / 1 for dry) output
Setup water pump

- The water pump is submersible, sucks in water from the whole and pumps it out from the little outlet.
- It needs to be connected to the board with the TIP31C transistor in order to switch it on and off.
Group everything together.

☐ System run 24/7. Every 15 minutes it checks the sensors in the following order and acts accordingly.
  ☐ Average last 10 readings from moisture sensor and get a percentage
  ☐ if soil humidity is less than 70% in pot plant, activate (turn on) water pump for 3 seconds then turn off.
source: I built an automated irrigation system, because i’m lazy
Information insights.

Blynk

Python telegram bot

Slack api

Using Python 3

Ubidots
End.

Special Thanks to

Mpho Mphego

🔗 https://twitter.com/mphomphego
🔗 https://github.com/mmphego

Follow me

🔗 https://twitter.com/AneleMakhaba
🔗 https://github.com/amakhaba
Questions