Justin Bland - 218478549
SIT210 - Task 11-2P

Task Objective

You are expected to submit all the artefacts of your system prototype, including source code, and a prototype report. The artefact is a complete version of your system prototype, and it needs to contain sufficient information so that someone else can use that to recreate your system prototype. The prototype report needs to outline your design approach, the architecture of your solution, and the working mechanism of the system. It also needs to present testing plans and any user manual to install and/or using your embedded solutions. A background and problem statement should be included for completeness, which can be modified from your project proposal.

You can use any template, but it must include the following sections:

- Overview, Including:
    - Background
    - Problem Statement
    - Requirements
- Design Principles
- Prototype Architecture
- Link to prototype code on GitHub
- Testing approach you have used for evaluating your system
- User Manual (this should be complete enough so that we can run your project on our own devices)
- Conclusion (should include 1-2 paragraph reflection on your experience, issues you faced, and how you would have done the project if given a second chance)

Task Submission Details

Provide a document outlining the section above. Your document should include the link to your GitHub and any other resources you have (e.g., in case you have videos)

GitHub Repository

https://github.com/CrashOverrideProductions/ProfessionalDevelopment/tree/main/Uni%20Files/SIT210%20-%20Embedded%20Systems%20Development/Task%2011.2P

Overview

Background & Problem

Owning and caring for reptiles can be a challenge, most reptiles have specific temperature, light and humidity requirements for good health, sometimes it can be difficult to know whether or not the environmental conditions inside the terrarium are healthy for a specific reptile.

Keeping the terrarium within the required values can be difficult at times additionally if issues arise with the heat lamps and UVB lamps. for example, a faulty UVB light will still emit visible light, just nothing in the UV spectrum and this will go un noticed, additionally with the use of Ceramic Heaters instead of heat lamps, if the Ceramic heater fails that too could go un noticed as they do not emit any light.

Requirements

This system has some specific requirements it needs to be able to measure the temperature and humidity of multiple points of the Reptile Terrarium, control the heating and lighting based on these values and user input variables.

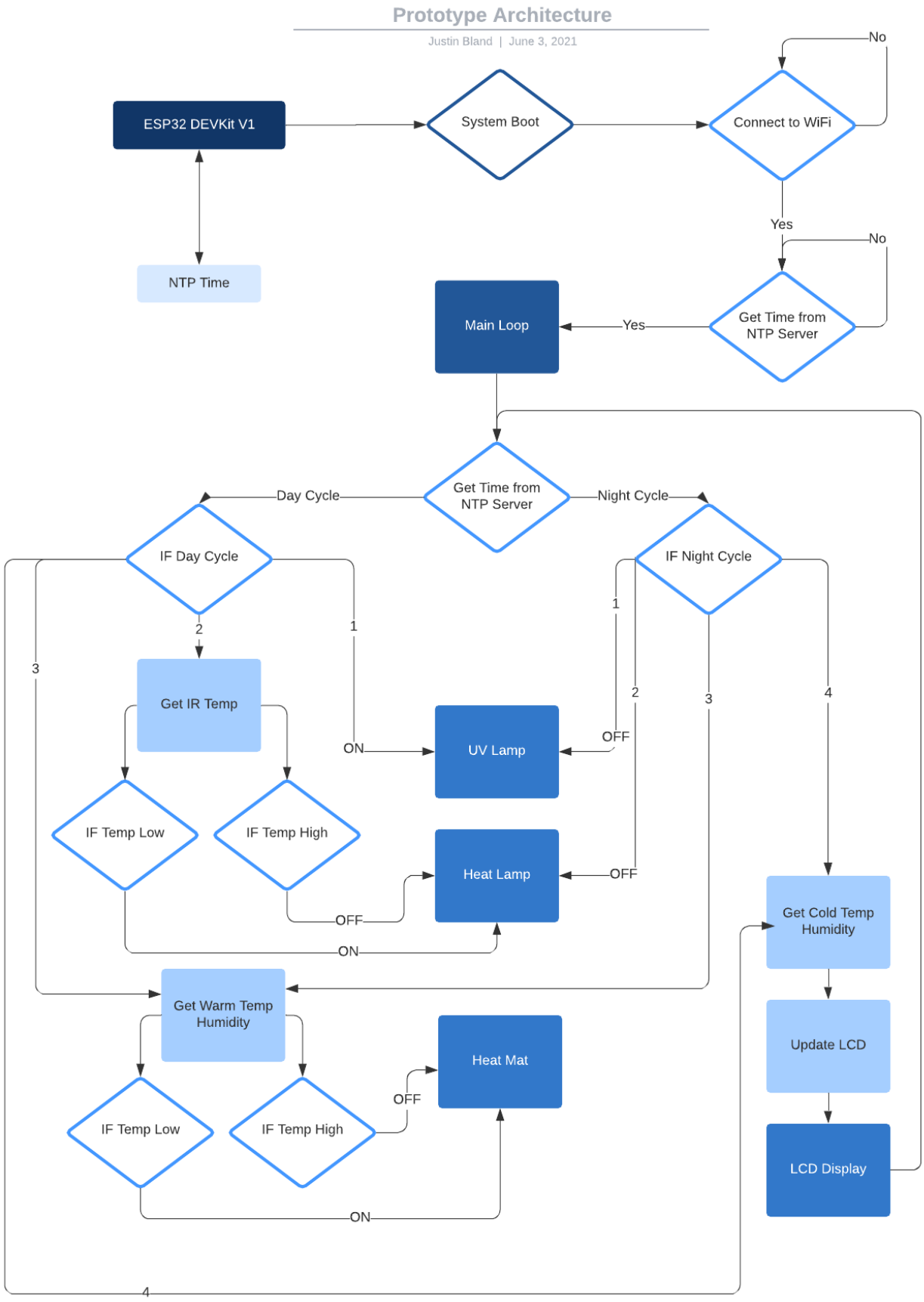
For this prototype the scope of reference will be limited to the following requirements

- Ambient Temperature and Humidity at either side of the Terrarium
- Temperature at the Basking Spot
- Control of UV Light, Heat Light and Heat Mat
- LCD Panel Data Display

Design Principles

- The system must connect to WiFi and a Valid NTP server prior to main loop start
- Once in the main loop the system will read all sensors and update the display
- The system will determine the time and day/night cycle and proceed with the correct cycle parameters
- For a Night Cycle
    - UV and Heat Lamps are turned off
    - Heat Mat is controlled to keep the warm side ambient temp within the specified range
    - All temperature and Humidity sensors are polled
    - Display is updated with updated values
- For a Day Cycle
    - UV Lamp is turned on
    - Heat Lamp is controlled to keep the basking spot within a specified range
    - Heat Mat is controlled to keep the warm side ambient temp within the specified range
    - All temperature and Humidity sensors are polled
    - Display is updated with updated values

Prototype Architecture



Link to prototype code on GitHub

GitHub Repository:

https://github.com/CrashOverrideProductions/ProfessionalDevelopment/tree/main/Uni%20Files/SIT210%20-%20Embedded%20Systems%20Development/Task%2011.2P

Testing approach you have used for evaluating your system

Testing and evaluating this system were approached by initially testing each individual component as it was written to ensure that each component was operating as expected prior to being included in the final prototype sketch.

Testing and evaluating the prototype sketch was approached by using short variables for the day night cycle in order to test that the system could switch between the two modes. Testing of the temperature monitoring and cycling was done using an external heat source, in this case an SMT Rework Station to warm up the area around the sensors to trigger a state change, this state change would display the current temp and or humidity on the LCD panel along with toggling the relays to turn on or off the relevant heat source, or in this case toggle an LED on or off.

User Manual

This prototype does not include the UV Light sensor as stock has been hard to source at the moment and for safety reasons does not include the relays for switching 240V mains power, this function is being emulated by 3 LEDs. With one additional substitution being the 2 DHT11 sensors listed in the Project Pitch being condensed into 1 sensor as the second decided to let the magic smoke out and stop working during development.

In order to run this project on you own you will need the following hardware

Microcontroller	DOIT ESP32 DevKit V1 (30-Pin)	Discontinued Early 2019
TFT-LCD	128x128 LCD Screen Module	https://www.jaycar.com.au/128x128-lcd-screen-module-for-arduino/p/XC4629
Infrared Temp Sensor	XC3704	https://www.jaycar.com.au/non-contact-ir-sensor-module/p/XC3704
Temperature & Humidity Sensor	DHT11	https://www.jaycar.com.au/arduino-compatible-temperature-and-humidity-sensor-module/p/XC4520
3x LEDs	Green 3mm LED 6500mcd Round Clear	https://www.jaycar.com.au/green-3mm-led-6500mcd-round-clear/p/ZD0124

Hardware Configuration

Using a bread board connect your components following the pin configuration below

ESP32 DevKit V1 Pin Configuration			
Device	*Pin*	*Pin*	*Device*
		36	LCD – SCK
		39	IR – SCL
		42	IR – SDA
DHT11 – Signal	16	35	LCD – SCK
UV Lamp LED +	17	34	LCD – CS
Heat Mat LED +	18	24	LCD – RST
Heat Lamp LED +	20	22	LCD – DC
Ground			Ground
			3.3V

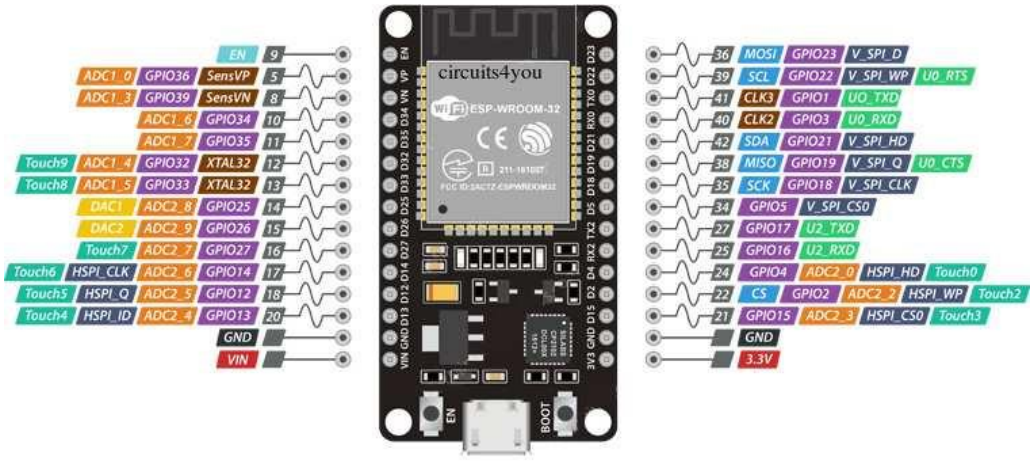


Image Credit: Circuits4You https://circuits4you.com/2018/12/31/esp32-devkit-esp32-wroom-gpio-pinout

Software Installation

The firmware for this project is developed using Visual Studio Code and PlatformIO, installation and configuration guides can be found in the links below.

Visual Studio Code:

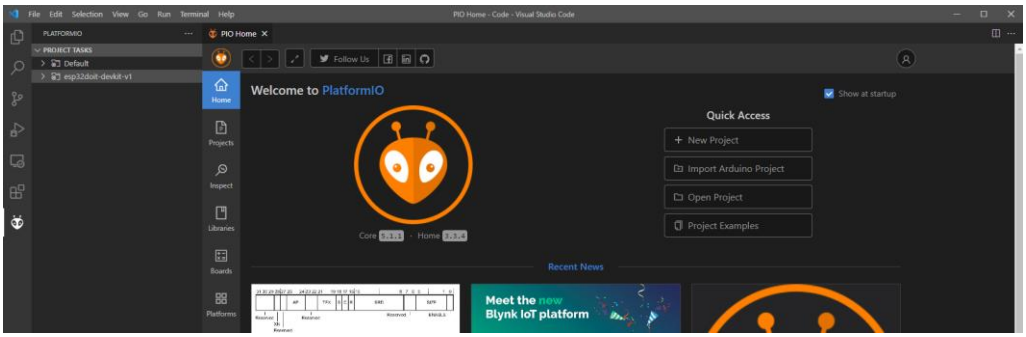
https://code.visualstudio.com/docs/setup/setup-overview

PlatformIO:

https://platformio.org/install

Firmware Upload:

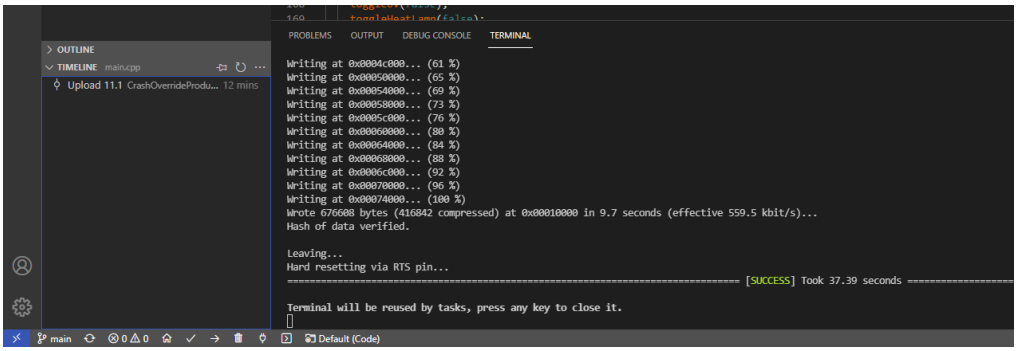
Download and extract the firmware (Firmware.zip) from the GitHub Repository to your local computer, next open Visual Studio Code and in the left menu click on PlatformIO plugin, this should open up the PlatformIO main screen, in the right-hand side click on Open Project, then proceed to navigate the folder where you extracted the firmware and click open.



There is one configuration change you will need to make, in WiFiFunctions.h on lines 6 and 7 you will need to substitute your own WiFi details. Without WiFi the firmware will halt as it uses a NTP server for time.

```
5
6   const char* ssid     = "UniFiNetwork";
7   const char* password = "14231315";
8
```

From here it is straight forward to compile and upload the firmware, plug in your ESP32 DevKit V1 into your computer and in PlatformIO in the bottom left click the right arrow and this will compile and upload the firmware to the device.



PlatformIO will download and install any Board Definitions, Libraries and Frameworks if needed, so an internet connection may be required the first time this firmware is compiled, if everything was successful you should be greeted with the terminal output above.

Conclusion

This project has been something I have wanted to do for a while now, with this unit giving me the opportunity to lay some ground work and hone in on its requirements and limitations, I will probably continue this project and use my existing work as a base for a new design, something along the lines of what was submitted in the Design PCB project. What I will definitely change is the temperature and humidity sensors to something a little more accurate and reliable than the DHT11.

I have learned 3 important things with this project so far, 1 SPI based devices are great for cutting down on GPIO requirements, the DHT11 sensor is problematic for reliability and accuracy and I need to spend time refactoring my code bases,