Documentation

Table of contents:

Description

We developed a plywood flower pot that supports automated water refill and light emission with the help of a water pump, container for water reserve and a LED-strip. The A display is attached to the pot that shows current levels of light, soil moisture and how much water is left in the container so that the user can see when to refill the tank. The same information is sent to the server with a Wifi-module. The power is supplied with the use of a power block, and this can be also done using a 12v rechargeable battery and voltage regulator, which is what we plan to do in the near future.

Other options on the market include pots that have self-watering systems, musical features, light supply and ability to tweet. Most of the contenders are rechargeable and offer a water reserve as well. Those that light the flowers have specific modes for different plants, which we do not have yet. What we have though is the choice of three colors that the LED-Strip can emit: red, green, blue, which the user can choose on the server.

We use the soil moisture detection sensor YL-69 to measure the level of soil moisture in the pot so that later we can change the level if the soil is too dry, using a pump. To maintain a constant level of lighting, in cases when lighting level is not enough for plant growth LED-light is automatically turned on. Also, it can be turned on intentionally from a webpage in 3 different lighting modes: red, green, blue.

We use the following features, communication interfaces that STM32F4 offers:

- *timers*
- *I^2C (for display)*
- *UART (for data transferring with the Wifi-module)*
- *ADC (for photoresistor and hygrometer)*

Sensors:

Hygrometer YL-69

- Vcc power supply - 3.3V or 5V
- Current - 35mA
- Signal output voltage - 0-4.2V
- Digital Outputs - 0 or 1
- Analog Resistance (Ω)
- Panel Dimension - 3.0cm by 1.6cm
- Probe Dimension - 6.0cm by 3.0cm

We use the soil moisture detection sensor YL-69 to measure the level of soil moisture in the pot so that later we can change the level if the soil is too dry, using a pump.

We receive an analog input from the pin A0 on the hygrometer and receive the data on ADC pin on STM in the range of 0-4095 (since our ADC on STM32F411 has 12bits resolution). As we measured, extremely wet soil gives 3100, so we normalize received information 0-3100 into 0-100.

LDR - Photoresistor

- Max power dissipation - 200mW
- Max voltage @ 0 lux - 200V
- Peak wavelength - 600nm
- Min. resistance @ 10lux - 1.8kΩ
- Max. resistance @ 10lux - 4.5kΩ
- Typ. resistance @ 100lux - 0.7kΩ
- Dark resistance after 1 sec - 0.03MΩ
- Dark resistance after 5 sec - 0.25MΩ

In order to get the level of the lighting, we use a photoresistor. Depending on the level of lighting, the photoresistor changes its resistance. The electric current passing through the photoresistor is transmitted to the STM32411DISCOVERY through the pin. Electrical voltage data is converted using ADC to binary code.

The minimum value we get (in the dark) is 0, the maximum is ~ 4 095.

When the photoresistor reaches the maximum required light for a plant, the value of the electric voltage is 1200. All the data that is greater than 1200, we accept as 1200. This data is normalized to a scale from 0 to 100.

Ultrasonic distance sensor HC-SR04

- Input Voltage: 5V
- Current Draw: 20mA (Max)
- Digital Output: 5V(High) , 0V (Low)
- Working Temperature: -15°C to 70°C
- Sensing Angle: 30° Cone, Angle of Effect: 15° Cone
- Ultrasonic Frequency: 40kHz, ±2kHz
- Range: 2cm - 400cm
- Accuracy - to 3 mm
- Dimensions:
    - Length 43mm Width 20mm Height 15mm
    - Centre screw hole distance: 40mm x 15mm
    - Screw hole diameter: 1mm (M1)
    - Transmitter diameter: 8mm

We use an ultrasonic distance sensor to measure the distance from the tank's surface to water and thus find out how much water is left in the tank.

We supply it with power and ground and then, the Trig should be a pulse (logical unit) for about 10 ms. It will start generating the locating signal. The signal is generated independently, Trig only launches it, so the exact duration is not very important. Having removed the Trig, you should start listening to Echo as soon as it rises to 1, and measure the time before the level drops to 0.

From the received time, distance to the obstacle can be found as $l = tc/2$, where l - distance to the obstacle in meters, t - the pulse duration on Echo, c - the speed of sound (the speed of sound in the air at a temperature of 20°C ≈ 343 m/s). The speed of light in the air can be found as follows: $c \approx 331.5+0.6 \cdot t$, where t - temperature in degrees Celsius. If the pulse lasts 38 seconds -- to the obstacle is far away, or the surface badly reflects the sound wave.

The microcontrollers record the duration of the impulse by interrupting the EXTI: the moment of change of the state of the ECHO line from 0 to 1, then - back, is fixed; the duration is the time difference between them. The timer, of course, is used - to measure time, but only as a counter.

For timing - both when presenting a signal to TRIG and for measuring the duration of ECHO, a timer from STM32F411 is used.

Actuators:

STM32F411Discovery

- 32-bit ARM Cortex-M4F core, 1 MB Flash, 192 KB RAM in an LQFP100 package
- Board power supply: through USB bus or from an external 5 V supply voltage
- External application power supply: 3 V and 5 V
- Peripherals: I2C, USART, UART, SPI, WWdG, IWDG, timers, ADC

We use the microcontroller to supply the moisture and light sensors with power, to maintain the Watchdog safety system, to receive data, then normalize and analyze them so that later we can set the pump or the LED-strip in motion and send the relevant data to the server and the display.

We use an ADC pin to receive the moisture level and the lighting level, as well as a UART pin to send the data to the ESP. Also, we use built-in timers for the LED-strip and the distance sensor. Via I2C pins, we communicate with the display LCM1602. We use the WWDG to maintain the watchdog.

Lolin NodeMcu V3 and ESP8266

- Supports Wi-Fi protocol IEEE 802.11 b/g/n;
- Integrated TCP/IP protocol stack.
- Supported modes of Wi-Fi – access point, client;
- Input voltage: 3,7B – 20 B;
- Operating Voltage: 3.3V
- Maximum current- 220мA;
- Built-in stack TCP/IP;
- The range of operating temperatures - from -40C to 125C;
- 32-bit processor;
- Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
- Digital I/O Pins (DIO): 16, 1 ADC Pin, 1 UARTs, 1 SPIs, 1 I2Cs
- Flash Memory: 4 MB, SRAM: 64 KB
- Clock Speed: 80 Mhz

The ESP board on LoLin is used to create a server, which enables us to control the color of LED-lighting via a webpage and display all the information about current state of the pot: moisture and lighting level as well as how much water is left in the tank. The module turns on, connects to the WIFI on a device with given login and password. Using our UART pins on STM32, we send the information to the module. The server starts and the user can access the webpage to either look at the data or change the color. The data is sent to the server 1000 times a second, so it is always relevant, and the PWM-generated slow changes in lighting can be executed.

LED-lighting SMD-5050

- Common anode RGB LED-strip
- Input voltage - 12V
- 60 LEDs per meter

To control the intensity of lighting we use the signal generated with PWM connected via three optocouplers. Colour of LEDs is controlled from ESP8266. In this case, optocouplers are used as switches, that close electric circuits responsible for required color.

To maintain a constant level of lighting, in cases when the lighting level is not enough for plant growth, LED-light is automatically turned on. Also, it can be turned on intentionally from a webpage in 3 different lighting modes: red, green, blue, which can also be mixed. The three GNDs on the strip turn on the corresponding colors.

Pump

- Input voltage - 5V
- 60mm x 28mm

To transmit signal onto the pump from STM32 we use an inverted relay. It turns the pump on, and after 5 seconds turns it off, when the opposite signal is sent to it. Then we wait for 6 seconds and remeasure the level of the soil. In the case if the pump has been working for more than 10 seconds, we turn the system off using the ATtiny13 safety system.

Display YWRobot LCM1602

I2C Address Range	0x20 to 0x27 (ours - 0x27)
Operating Voltage	5 Vdc
Backlight	White
Contrast	Adjustable by potentiometer on I2C
Size	80mm x 36mm x 20mm
Viewable area	66mm x 16mm

We used YWRobot LCM1602 IIC V1 display to show the measurements of the lighting and the soil moisture level in real time. In order to connect the display via I2C to the microcontroller, we used a modified version of this **library**. We connected two pins of the display, SDA (I2C data line) and SCL (I2C clock) to two pins on the microcontroller, and transferred data about water supply, soil moisture, and light level onto the screen.

Safety System

ATtiny13

- 8-bit AVR Microcontroller with 1K Bytes In-System Programmable Flash.
- Operating Voltage: 2.7 - 5.5V

We use ATtiny13 for technical security. One of the possible scenarios of an accident is the case when the pump works for too long, pouring the water onto the soil. This poses the danger of overpowering it and in severe cases, flooding the actual pot, and thus the plant may suffer.

In order to prevent this, we first check whether the pump is supplied with power (and thus, working and pouring water), using an optocoupler. With an inverted signal from that optocoupler, we check the voltage that goes to the pump and put 0 or 1 onto the pin at ATtiny13. The ATtiny is programmed to count whether that signal has been 1 for at least 10 seconds, and then another pin outputs 0. This signal is sent to 3 relays, thus controlling the supply of power, and breaking the electric circuit in case it is 0. Those relays stop the power flow to the LED, pump and the rest of the system.

You can read the datasheet for ATtiny13 here.

Watchdog Timer

Watchdog Timer is a guarding timer that represents a hardware-implemented control system against the project's system freezes. This is a timer that periodically updates (drops to 0) its value in the system, and in the case when this update was not completed and the timer "counted" to its maximum, the microcontroller reboots.

Specifically, in our system, it may be useful in case of a "freeze" (which in our case is - waiting for an answer for too long) of the I2C data bus, which provides the communication with the display. Or if for a short period this bus is somehow disconnected (it is likely that the crash will occur and the system will have a forced reboot). There is the Independent Watchdog (IWDG) and Window Watchdog (WWDG) - the difference is that the second has two limits (upper and lower) and if the update is faster / slower than necessary, the system will be forcefully restarted. Since the IWDG is clocked from its own pulse clock generator, it will work even when the source of the controller tact will fail.

When setting the timer update period, you should look at the clock frequency of its generator, for example, at a clock frequency of 32 KHz and prescaler equal to 32, we find that with the period of 1000, the timer will reset and restart the controller in 1 second. The timer update is carried out by the command: HAL_IWDG_Refresh (& hiwdg).