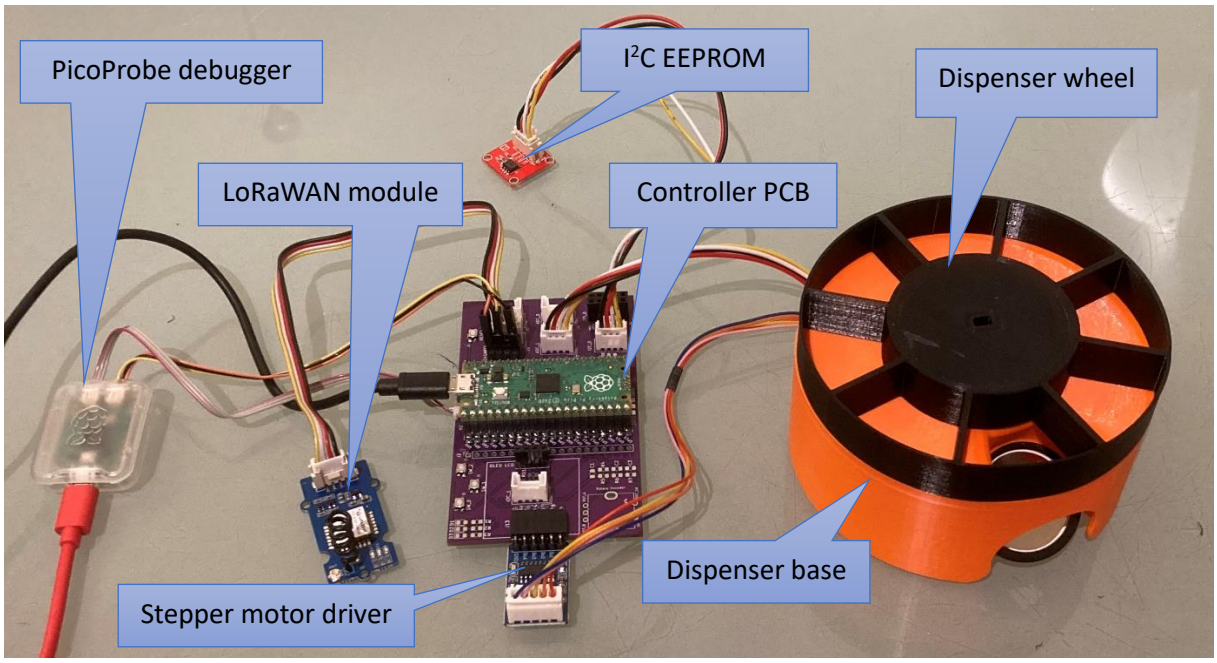
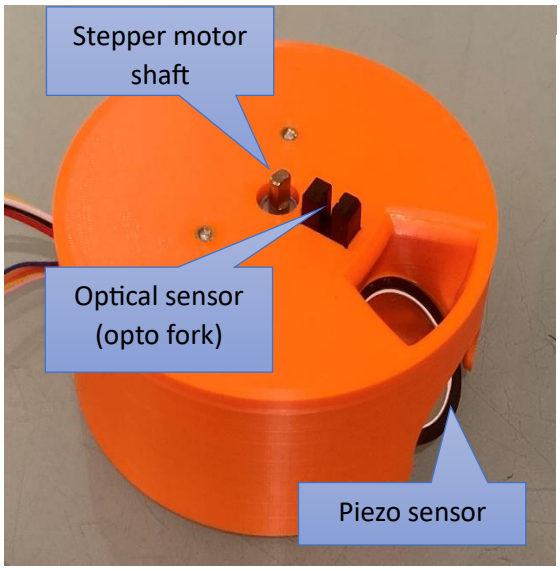
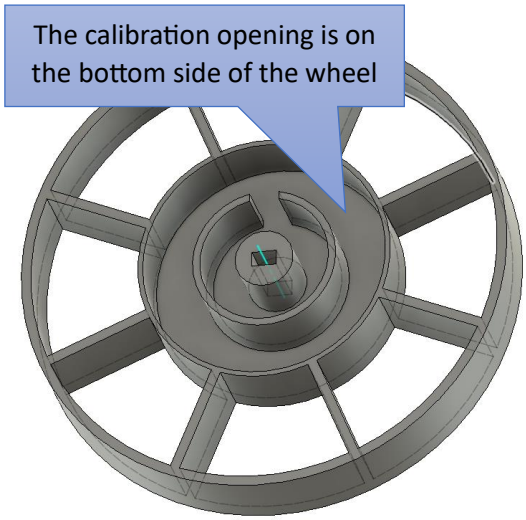
Pill dispenser

Pill dispenser consist of three main parts: controller PCB, dispenser base and dispenser wheel. Dispenser base contains a stepper motor for turning the dispenser wheel, a piezo electric sensor for detecting if a pill is dispensed and an optical sensor for calibrating the wheel position. Dispenser wheel has eight compartments, seven for pills and one for calibration. The wheel has no sensors but has an opening that can detected by the optical sensor of the base to calibrate the wheel position.







Status of the device can be communicated to the server using a LoRaWAN module. LoRaWAN is a long-range low bandwidth radio network standard for transmitting relatively small amounts of data at a time. LoRaWAN can be setup as a private network of bough as a service from operators.

Wiring and pin assignments

Dispenser base has a 6-pin JST connector and a 4-pin Grove connector. Connect the JST connector to the 6-pin connector on the stepper driver board. There is only one 6-pin connector in the system so there is no risk of incorrect wiring. There are multiple grove connectors on the PCB. Connect the Grove connector from dispenser base to ADC_1 connector.

LoRaWAN module is connected to UART0 connector under the stepper motor driver.

Dispenser pin assignments:

- Opto fork – GP28 – Configure as an input with pull-up
- Piezo sensor – GP27 – Configure as an input with pull-up
- Stepper motor controller – GP2, GP3, GP6 and GP13
    - All four pins are outputs
    - Pins are connected to the stepper motor driver pins IN1 – IN4
- LoRaWAN – uses UART, can be connected to UART1 or UART0
    - UART0 uses GP0 and GP1
        - Note that UART0 is used by stdin/stdout when UART stdio is enabled so **LoRaWAN module should be connected to UART1**
    - UART1 uses GP4 and GP5
- LEDs
    - GP20 - GP22

Important information

Never turn the dispenser wheel by hand!

Project goals

The goal of the project is to implement an automated pill dispenser that dispenses daily medicine to a patient. The dispenser has eight compartments. One of them is used for calibrating the dispenser wheel position leaving the other seven for pills. Dispenser wheel will turn daily at a fixed time to dispense the daily medicine and the piezoelectric sensor will be used to confirm that pill(s) were dispensed. The dispenser state is stored in non-volatile memory so that the state of the device (number of pills left, daily dispensing log, etc) will persist over restarts. The state of the device is communicated to a server using LoRaWAN network.

For testing purposes, the time between dispensing pills is reduced to 30 seconds.

Minimum requirements

When the dispenser is started it waits for a button to be pressed and blinks an LED while waiting. When the button is pressed the dispenser is calibrated by turning the wheel at least one full turn and then stopping the wheel so that the compartment with the sensor opening is aligned with the drop tube.

After calibration the program keeps the LED on until user presses another button. Then it dispenses pills every 30 seconds starting from the button press. When the wheel is turned, the piezo sensor is used to detect if a pill was dispensed or not. If no pill was detected the LED blinks 5 times. When all pills have been dispensed (wheel turned 7 times) the program starts over (wait for calibration, etc.).

Advanced requirements

Device remembers the state of the device across boot/power down.

Device connect to LoRaWAN and reports status of the device when there is a status change (boot, pill dispensed/not dispensed, dispenser empty, power off during turning, etc.)

Device can detect if it was reset / powered off while motor was turning.

Device can recalibrate automatically after power off during middle of a turn without dispensing the pills that still remain.

Lora communication

Lora commands for connecting to the network:

1. +MODE=LWOTAA
2. +KEY=APPKEY,”<hexadecimal key for your device>”
3. +CLASS=A
4. +PORT=8
5. +JOIN
6. +MSG=”text message” (Can be used only after successful join)

You can receive and display the messages that you send by using lorareceive.py from Documents/project. The program has short instructions in comments and the beginning of the source code.