DailyDose: Smart Pill Dispenser

by chloedevriese

Welcome to my project called DailyDose!

My name is Chloë Devriese, I'm a student Multimedia and Communication Technology at Howest in Kortrijk, Belgium. As an assignment for school, we needed to make an IoT-device.

When visiting my grandpa, I got the idea for my project. My grandpa needs to take a lot of medications during the day but It's not always easy for him to take the right pills at the right time. It can sometimes be too confusing for him. Yet it is important that the correct amount of medication is taken at the right time. To make this easier for my grandpa and for many people, I came up with the idea of DailyDose.

DailyDose will tell you exactly when and which medications you need to take. When It's time to take a medication, the alarm will go off. The only thing the patient has to do is push the button and the right medications will come out of the dispenser.

A doctor or loved one can fill up the medications by removing the top of the dispenser.

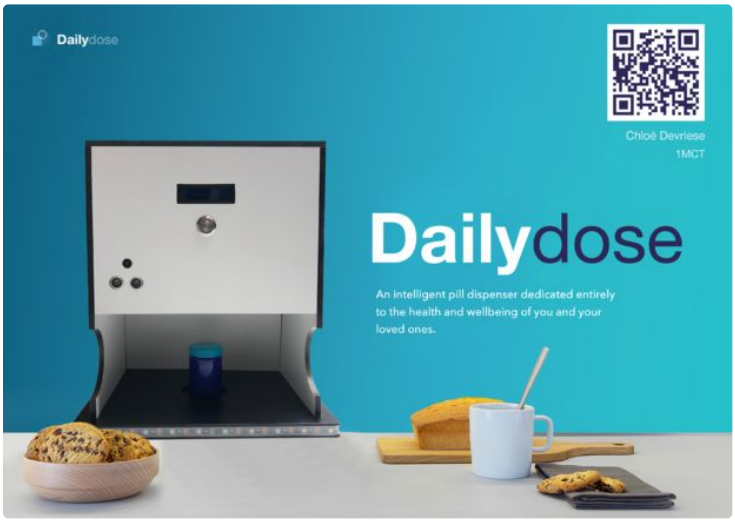
4 containers for 4 different medications are present in this prototype.

The temperature inside the dispenser is also regularly checked. The reason for this is that

pills need to be stored at a temperature below 25°C otherwise they can become toxic.

Next to the construction, I made a website to control the dispenser. You can give in more information about the patient and his medications. Besides that, you can generate the dose schedules.

Below you can find an explanation of how to make DailyDose. If you want to know more about me and my other projects, check my portfolio.



Step 1: Collecting the Materials

First things first, I needed to make sure that I had all the required parts. Before we begin I would like to say that this project wasn't exactly cheap. Below you can find a list of the different components that I used. I also included a bill of material with all the prices I paid and possible retailers for the components.

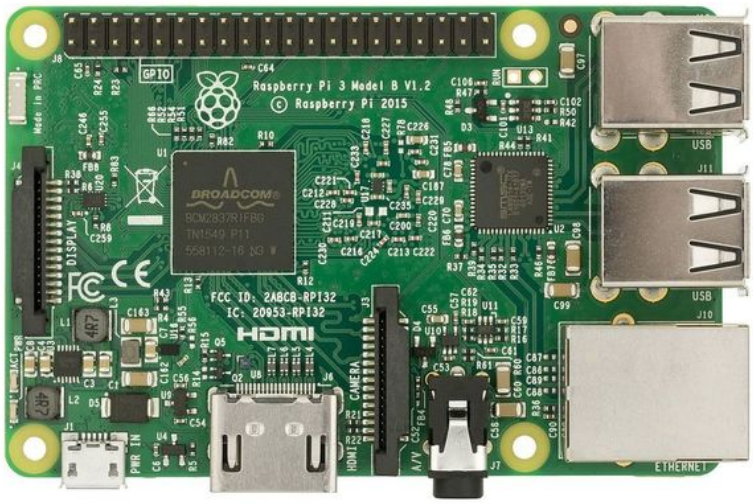
- RaspBerry Pi 3 with adaptor and memory card
- Jumper Cables
- Breadboard(s)
- 1x 4,7K ■ resistor
- 1x 3,3K ■ resistor
- 2x 470K ■ resistor
- 1x 1K ■ resistor
- LCD Display
- DS18B20 One Wire Temperature Sensor
- Square Force-Sensitive Resistor (FSR)
- Mcp3008*
- Ultrasonic sensor
- 4 x continuous rotation servo motor (FS5106R)
- Button**
- NeoPixel rgb LED Strip (30 LED- black)
- Logic level converter ***
- Power Jack
- 5V/2A DC power supply ***
- Active Buzzer

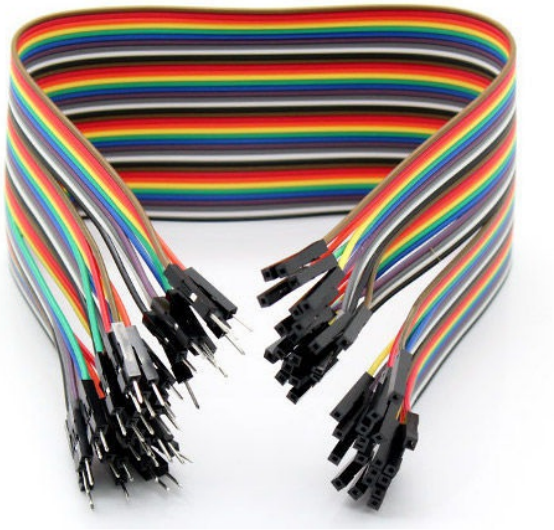
Notes:

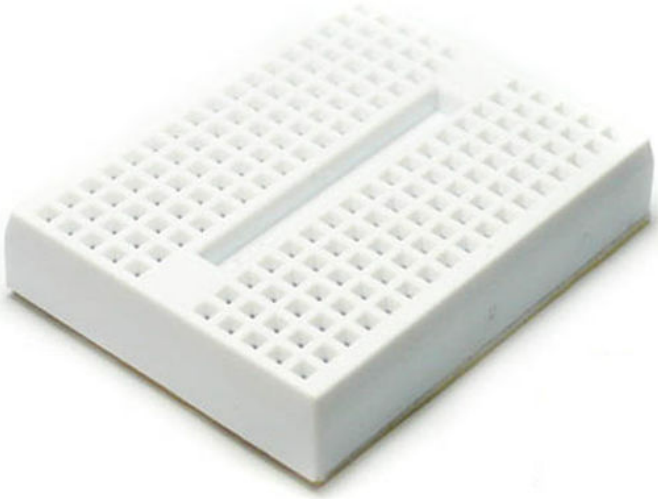
*The raspberry Pi doesn’t have analog input pins. To solve this problem, I used a mcp3008 to convert an analog signal to a digital signal.

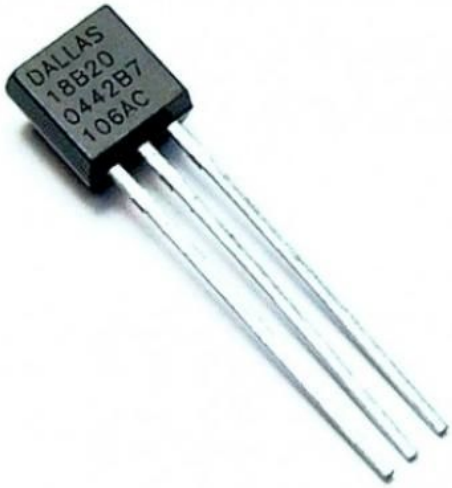
**I used a Rugged Metal RGB Pushbutton, but you can use any button you like. I picked this button because first of all not gonna lie, it looked pretty cool. It’s also a button that stands out. Because my target audience is mainly elderly, it had to be a button that's clearly visible.

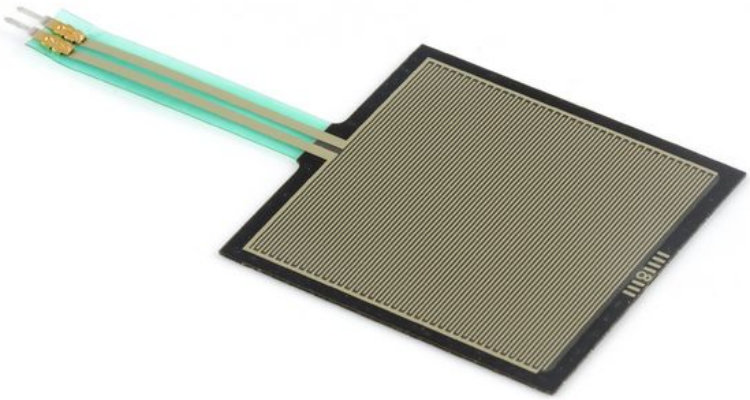
***The Raspberry Pi uses 3.3V Logic, so we will need to use a Logic Level Converter to convert it to the 5V logic that Neopixels require. You will need to use an external power source, as NeoPixels take a LOT of power. Each pixel will draw about 20mA on average, and 60mA at white - max brightness. 30 Pixels will draw 600mA on average, and up to 1.8A. Ensure your power supply is large enough to drive your strip!

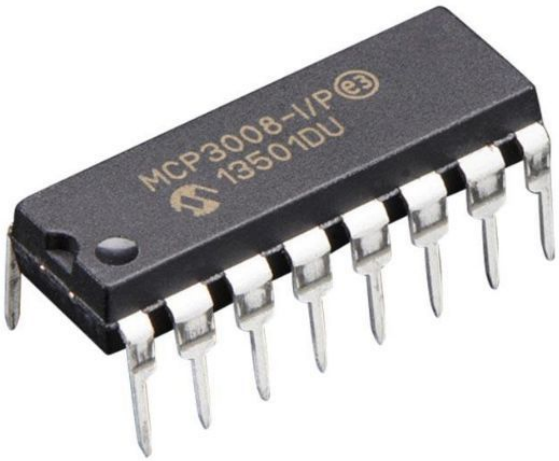

Raspberry Pi 3 Model B V1.2
© Raspberry Pi 2015
BROADCOM
FCC ID: 2ABCB-RPI32
IC: 20953-RPI32
HDMI
CAMERA
DISPLAY
USB
ETHERNET





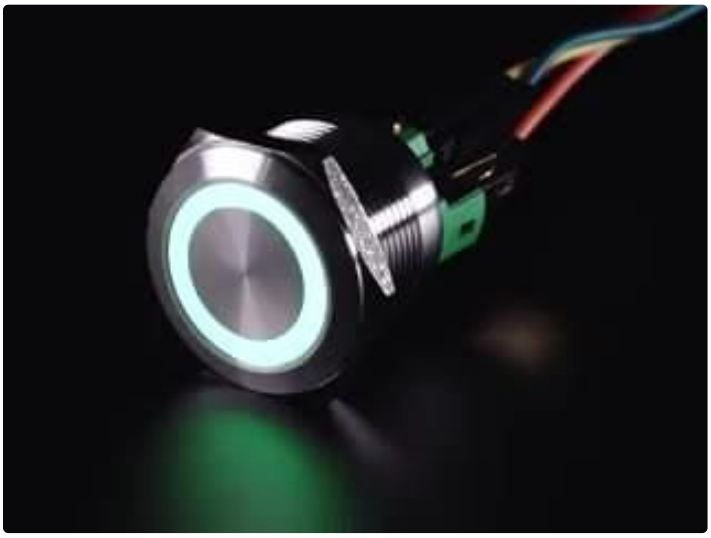

DALLAS
18B20
0442B7
106AC

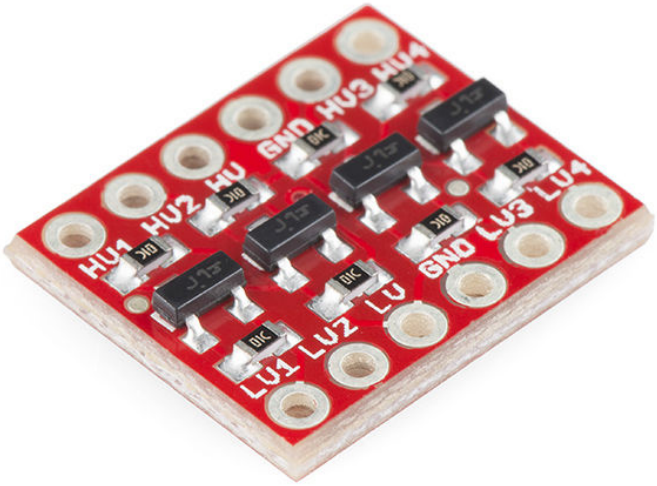



MCP3008-I/P
1350 1DU


Y1
HC-SR04
Vcc
Trig
Echo
GND
T
J1
R


FITEC
FS5106R



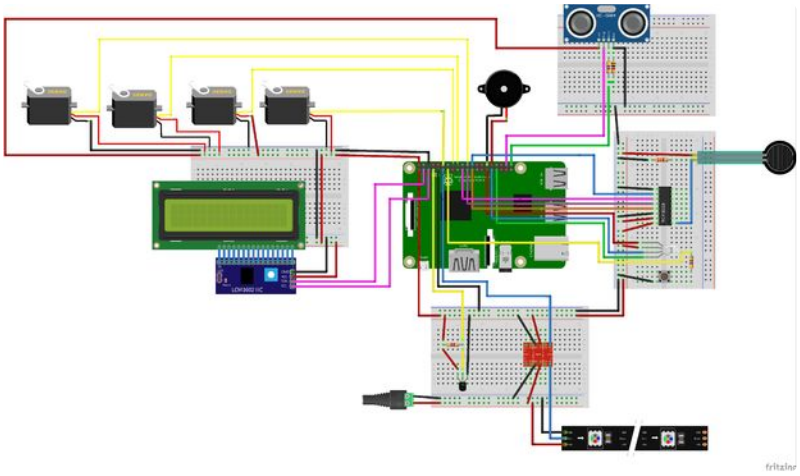


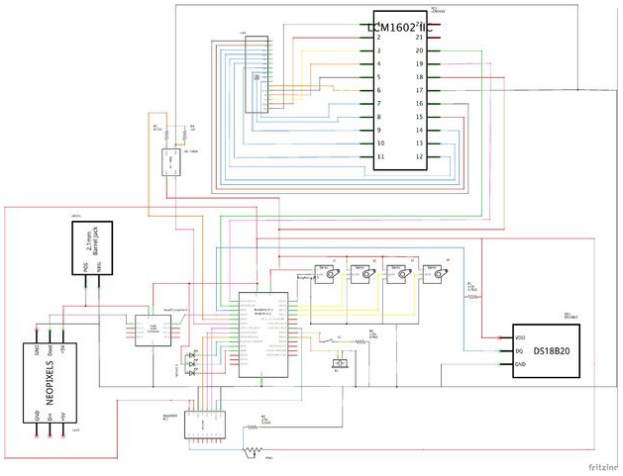
https://www.instructabl...

Download

Step 2: Wire Everything Up

On the picture you can see how to build the circuit. It is actually not that difficult. I couldn't find a Rugged Metal RGB Pushbutton so in the schematic circuit I used a regular button and an RGB common anode led to represent the lights in the button.





Step 3: Database

For this project we need a database.

I created an entity relationship diagram, made a database of it and inserted some testdata. Soon it was clear there were some mistakes, so I did it again and again. Later when I started programming, I discovered that there are still some little issues with the database but for this prototype it did the job.

The table SensorHistory has information about the sensors. It captures the measured temperature in the dispenser, it checks whether a cup is present underneath the dispenser so that the pills don't just fall down into nothing. It also checks how far away the patient is when the alarm goes off.

You can use the dispenser for one patient. The information about this patient is stored in the table patient.

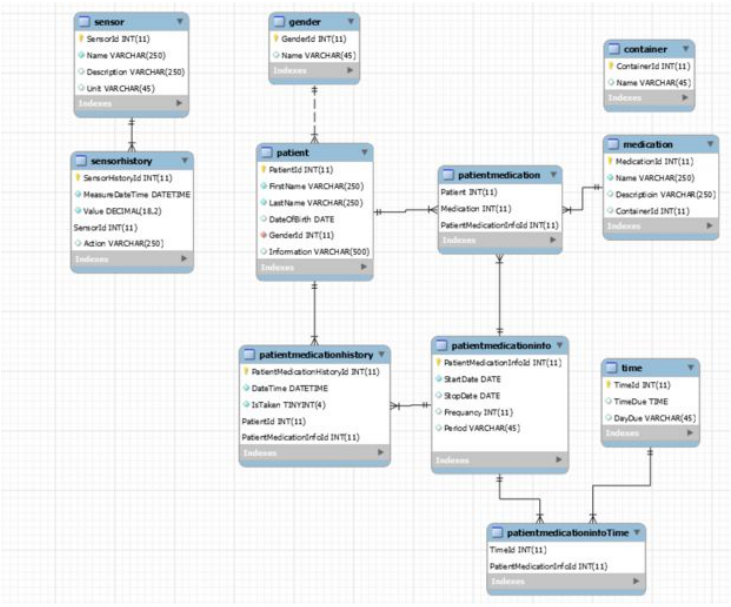
Any medication you want, can be added to the medication table. You can also add a medication that's not stored in a container.

With the tables PatientMedication , PatientMedicationInfo, PatientMedicationInfoTime and Time we keep track of the dose schedules of the patient.

The PatientMedicationHistory keeps track of whether the patient has taken his medications at the right time, yes or no.

Attached to this step you can find my Mysql dump. So, you can easily import it.

Now that you have the database it’s time to set up your RPI and implement the database.



https://www.instructabl...

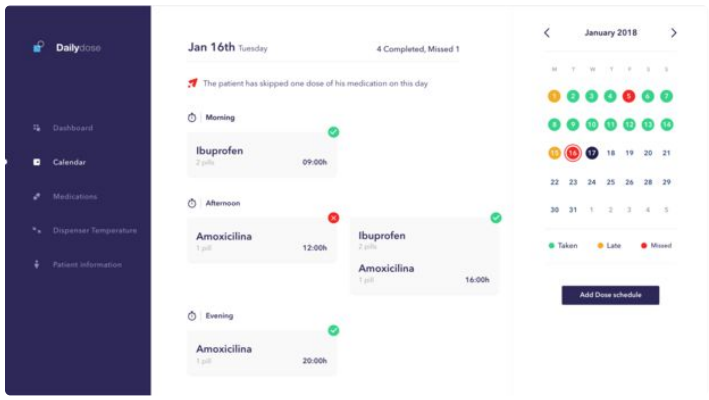
Download

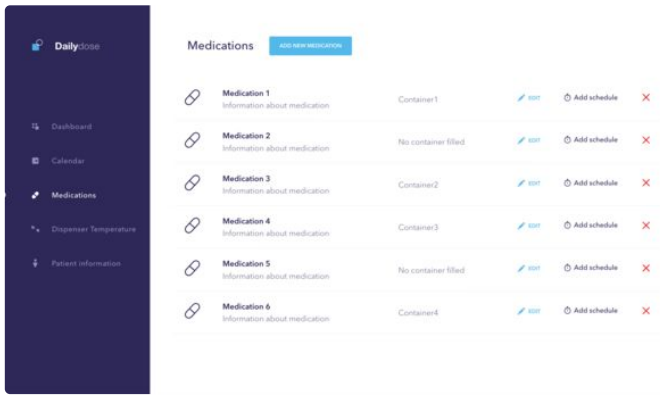
Step 4: Code It!

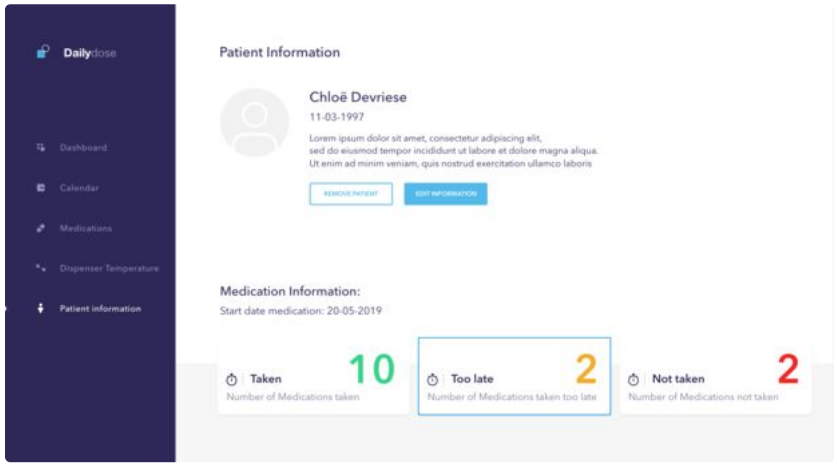
Now it's time to make sure all components do their job. You can find my code on Github.

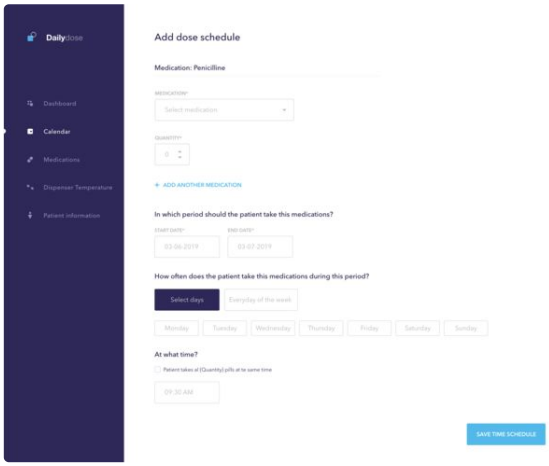
https://github.com

Download the code









Step 5: Building the Dispenser

For the dispenser I used multiple HPL plates and one plate of MDF

The construction

HPL:

2 x - 35cm x 25cm (left and right side)

1 x – 35cm x 28cm (back)

1 x – 21cm x 28cm (front)

2 x – 23cm x 28cm (middle support and small part of lid)

1 x – 25cm x 30xm (big part of lid)

In the HPL plate of 21cm x 28cm (front) you provide openings for the components (Lcd, button, ultrasonic sensor and buzzer)

In the back and middle support plate you provide a hole for the power supplies. You also provide a hole in the middle of the support plate so the pills can fall down

MDF:

1x – 30cm x 27cm x 2cm (bottom part)

Provide a notch in the MDF plate, all around, with a height of 1,2 cm. This is necessary for the LED strip.

In the middle of the plate you make a round notch with a small hole to the backside off the plate. This round notch is used to place a cup and the Force-Sensitive Resistor. The small hole is to hide away the cables of the Force-Sensitive Resistor.

If you want, you can now paint the MDF plate, this plate will be the bottom part.

When you have all the plates, you can put them together. I used teck7 glue. But be careful this is a tricky part you might need some help.

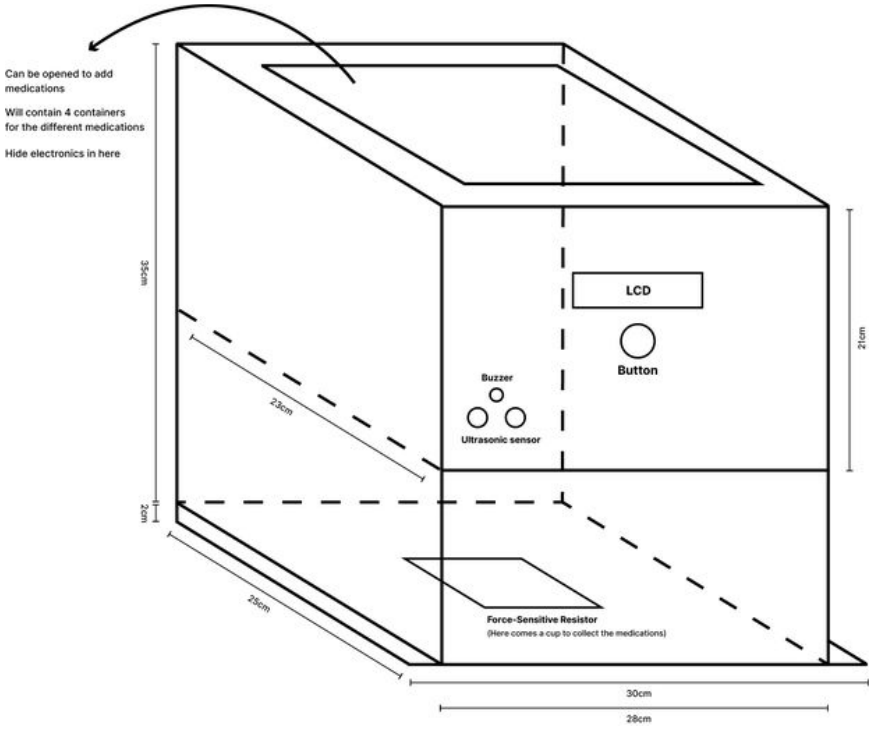
Some kind of funnel

U need a funnel so that the pills that come out of the container will fall into the hole in the middle support plate.

I made my funnel with cardboard, tape and glue. This was mainly by feeling.

Printing the 3D elements
I used 3D elements for the 4 containers every container consists of a cup, a servo rotator and a cup rotator

Download 3D-files


Can be opened to add medications
Will contain 4 containers for the different medications
Hide electronics in here
35cm
2cm
23cm
25cm
LCD
Button
Buzzer
Ultrasonic sensor
21cm
Force-Sensitive Resistor
(Here comes a cup to collect the medications)
30cm
28cm

https://www.instructabl…

View in 3D Download

https://www.instructabl…

View in 3D Download

https://www.instructabl…

View in 3D Download