
Table of contents

[Projects](#)

[EatThat](#)

[General description of the project](#)

[My involvement in the project](#)

[Laser Projector](#)

[General description of the project](#)

[My involvement in the project](#)

[Resource management mobile app based on QR codes](#)

[General description of the project](#)

[Entries](#)

[06/05/2014 - EatThat](#)

[07/05/2014 - EatThat](#)

[08/05/2014 - EatThat](#)

[09/05/2014 - EatThat](#)

[12/05/2014 - EatThat](#)

[13/05/2014 - EatThat](#)

[14/05/2014 - EatThat](#)

[15/05/2014 - EatThat](#)

[16/05/2014 - EatThat](#)

[20/05/2014 - EatThat](#)

[21/05/2014 - EatThat](#)

[22/05/2014 - EatThat](#)

[28/05/2014 - Laser Projector](#)

[29/05/2014 - Laser Projector](#)

[02/06/2014 and 03/06/2014 - Laser Projector](#)

[04/06/2014 - Laser Projector](#)

[05/06/2014 - Laser Projector](#)

[06/06/2014 - Laser Projector](#)

[11 to 18/06/2014 - QR resource management](#)

Projects

EatThat

General description of the project

please enter general description here

All software files are [in Dropbox](#)

Please include diagram of the circuit and pictures of the electronics prototype

My involvement in the project

Please enter info here - what you were asked to do and what you achieved.

Laser Projector

General description of the project

Need some info here

[Open project folder](#)

My involvement in the project

need info here

Please provide link to code, diagrams and pictures.

Resource management mobile app based on QR codes

General description of the project

Tibi initiated a project to create a resource management mobile app based on QR codes.

Fernando is working on inventory, placing QR codes on physical resources used by the [SENSORICA network](#). The QR codes point to a webpage on [SENSORICA's NRP system](#). See [THIS](#) video presentation of the concept.

Requirements

- user friendly
- authenticate the user to provide access to SENSORICA's NRP
- extract data from the NRP for specific resources identified by their QR code
- allow the user to modify data in NRP for specific resources identified by their QR code
- ...

Other ideas

Adapt behaviour according to the properties of resources. Examples:

- for **consumables** the user can: log use, change location, replenish.
- for **equipment** the user can: log use, schedule use, change location, log maintenance contribution
- for **space**: log use, log maintenance contribution

Questions and possibilities:

1. *Will the phone app itself want to post information to NRP?*
2. *Or should we think about simple forms that could be used on the phone's Web browser to post information?*

In case 1, the phone app will need to do the authentication and authorization with NRP. And get the required context information for logging use, changing location, etc. For use, the context information would be most easily provided by a commitment (planned use for a project), which NRP could give to the phone app. For changing locations, the phone app would also need to get the list of locations.

In case 2, the phone app could take you to the page, you could log in normally, and post the information, and the phone app would not need to do much. (Also, for case 2, the existing forms are too big for a phone; we'd need to make smaller and simpler ones. Not much of a problem, though.)

We are also creating QR codes for projects, but these ones point to the project's page on SENSORICA website. We can think of linking it to the VAS too.

My involvement in the project

Build a mobile app with a user-friendly interface to allow users to interact with SENSORICA's NRP and manipulate data about physical resources in the inventory.

[Link to Github repository.](#)

Entries

06/05/2014 - EatThat

- Today I worked with Python to build a script that connects to the Streaming API from twitter and keep tracking tweets with specific tags. To do this, I used the Twython API and learned about Oauth level 1 and 2 authentications.
- After doing this, I built an GUI listing the users and tweets filtered through the Streaming API. I also learned and used communication through serial ports, to send the tweets from my computer to the raspberry pi device, and vice versa.

07/05/2014 - EatThat

- Delved deeper into the Streaming API and used it to send tweets, direct messages and identify retweets. Also solved the problem regarding the communication through the serial ports. Used an oscilloscope to help me do it.
- Started programming to a microcontroller (LPC 810). It will control the machine that will serve the pasta on the event. Downloaded and started using LPCXpresso, lpc21isp and flashmagic, tools that I'll be using to develop for this chip. Managed to do an hello world application and upload it to the memory of lpc810. Starting to understand things better.

08/05/2014 - EatThat

- Studied a little more about the IDE that I'll be using for developing. Compiled and tested some sample projects, learned how to debug and gain access to the microcontroller API.
- Learned about how to configure and initialize some code for the lpc810. Also, managed to send a string through serial communication using UART.

09/05/2014 - EatThat

- Spent the day trying to figure out how to receive data through serial communication using the lpc810. The receiving is working, but I'm having some difficulties to interpret and gather the data in the software part.
- Learned a little about interrupts and how they work. Started reading the User Manual available at the nxp website and it is helping tremendously to understand better how the programming for the lpc810 is done.

12/05/2014 - EatThat

- Jonathan found a library contained in the ROM of the lpc810 that contains a lot of functions already built-in. One of them will help solve the problem regarding the data

receiving using UART0. Now I'm trying to find a way to access these libraries, searching on google and looking through forums.

- Victory! Managed to receive and interpret the data in the code using interrupts. Now I can do everything with UART, send, receive, interpret data. We should be ready to move to the next step now, which I think is controlling the machines to serve the alphabet pasta. I'm waiting now for Jonathan or Antonio to arrive and provide guidance regarding what to do now.
- Started to study the I2C protocol that I'll need to use to control the pasta dispenser machine. We need 26 pins to make it work, but the microcontroller doesn't have it. We will use a GPIO Expander controlled using the above mentioned protocol. This way, it is possible to control the wheel with just one controller.

13/05/2014 - EatThat

- Spent the day studying the I2C protocol and trying to code a function to read and receive data through it using the LPC810. I think the routine is read, but the chip don't have the memory to store the code. Jonathan ordered a new board with more space, which is going to arrive tomorrow.
- Discussed how the pasta dispenser is going to work on the hardware and software side. This way, I can start to work on it by myself if I arrive early and Jonathan and Antonio aren't here yet.

14/05/2014 - EatThat

- A new board arrived today, the LPC-810 MAX, which supports mbed. Because mbed is higher-level than vanilla C with LPCxpresso, I switched to it and got some awesome results. The communication through I2C is dealt with, and I even managed to work with the interrupt port of the chip. During the day, I've done some tests to learn the libraries, and by the end of it everything was ready and now the assembling of the Eat That project can start!

15/05/2014 - EatThat

- Today I finished the code that will run with the 26 motors, and used leds to test it, since we didn't have all the motors yet. I couldn't test everything, because the board that I'm using needs a modification so that I can have access to a virtual COM port. I asked Jonathan to do this for me, since I don't have knowledge in electronics, and probably everything will be finished tomorrow.

- The test with one motor went very well in my opinion. The response time is good, and most of the time only one piece of pasta falls, which is how it should be working. When it doesn't work, letting two or more pieces of pasta fall, is due to one being "bonded" to another, which is not our or the device fault.

16/05/2014 - EatThat

- Started to research how a step motor works because one will be used to control a wheel in C2MTL while the pasta is being dispensed. Tried to code something to make it work using the LPC800-Max board but didn't get success. Jonathan and Serge are looking at the motor wiring to make sure everything is all right.
- Looked for some GUI libraries from python. It is interesting that the Eat That app have a graphical interface, making it easier for people to operate it. The idea is that the user be able to modify all the configuration parameters through the interface, without the need to access the code. Furthermore, the interface should also show the tweets, classifying them according to some parameters that we are still thinking of.

20/05/2014 - EatThat

- Was able to make the step motor work with the help of Antonio. Now we are running some tests to see what the best speeds and accelerations would be for the wheel. We are using 16 pulses per step, which brings a lot of precision to the motor, so that's good!
- Started to integrate the step motor controlling code with the pasta dispensing one.

21/05/2014 - EatThat

- I'm working with Raspberry PI again, since this is the device that will keep receiving tweets from the Streaming API, filter them, and send them through the serial communication to the LPC board, which then will control the pasta dispensing. I'm mainly testing code, since most part of what should be the final version is finished. Since the code was written in Python and PI is able to run a Python interpreter, no porting was needed.
- Run into some problems regarding the serial communication. Somehow, that data that is being sent to the board from PI is being "messed", and I don't know, since the same code with the same connections worked before. The most strange thing is that, if I change the wiring to use the second UART module in the board, everything works.

Unfortunately I can't do that, since it is the same module that controls the I2C protocol, which I'm using to control the pasta dispensing motors...

- Found a solution for the problem stated above. I'm now using an FTDI cable to do serial communication and everything is working.
- After running some tests we found out that our initial plans, which was to use the LPC800-Max board to control both the pasta dispensing motors and the step motors at the same time is not going to work, since they are interrupting with each other, since each one needs some computation power. The board is not capable of multi-threading, so I wrote code to control the step motor through the PI, using a GPIO library for Python. All code snippets are ready and tested, so I'll start doing the final integration tomorrow.

22/05/2014 - EatThat

- Merged all the python code snippets together in the PI and the C code in the board, and started doing the final integration between the Streaming Twitter API, the Raspberry PI and the LPC800-max.

28/05/2014 - Laser Projector

- Today I started working with a temperature sensor. Managed to make it work on the Arduino, and now I'm trying to work with the lpc810.

29/05/2014 - Laser Projector

- Kept working with the temperature sensor and lpc810. However, couldn't make them work yet. There's a lot more code needed to program the lpc, since it's code is not nearly encapsulated as arduino or mbed.

02/06/2014 and 03/06/2014 - Laser Projector

- Still trying to make the sensor work with the lpc810.

04/06/2014 - Laser Projector

- Finally made the temperature sensor work. All the past components that I worked with here at Sensorica didn't need that commands were issued to make a reading. This sensor does need some procedures before a reading can be made, and after understanding this I was able to use it correctly.

05/06/2014 - Laser Projector

- Studied about PID and PWD. I'll need to use both to control a cooling system. The idea is to implement a PID control loop that is going to watch the temperature of a system using a DS18B20 (the temperature sensor I've made work before) and, based on the temperature of the system, will determine how long the duty cycle of the cooling system should be.
- Managed to implement the mentioned loop control above using mbed and the lpc810-MAX.

06/06/2014 - Laser Projector

- Done some testing of the system with Antonio.
- Studied and worked a little with Pixy. It'll be used in a future project that I'm going to work.

Missing some dates...

11 to 18/06/2014 - QR resource management

- communicated with Tibi and other sensorica members that work on OVNi to learn more about the project
- created github account
-