Monitoring the status of the garbage containers

Preliminary analysis

Authors: Xavier Cerqueda Puig

Bernat Garcia Torrentsgeneros

Pierre Biojoux

Quentin Studeny

Junyoung Bang

Joonas Luukkanen

Supervisor: Torben Gregersen

Date: 30/09/2016

1. Table of contents

2. Introduction

This document describes the pre-analysis carried out in order to implement the project "Monitoring the status of garbage containers".

For the design and implementation of this project, it is necessary to conduct a preliminary study in order to compare the different options available on the market.

In our case, we need a sensor to detect the status of garbage containers and a microcontroller to store data and control the process. In addition, also requires a communication module to emit the signal over long distances and to know the GPS location.

So basically this document is to analyse the selection of the different components that we are going to use for our prototype.

3. Microcontroller selection

3.1 Comparison of options

For possible microcontrollers we were looking for popular development boards with possibility to insert sensors. We chose not to opt for more uncommon platforms due documentation and help available for more popular ones. In table 1 we compare 4 different platforms from development perspective

Table 1 Platforms compared.

Platform	Arduino Mega2560	Arduino Uno	Raspberry Pi ZERO	Raspberry Pi 3 Model B
Microcontroller/ Processor	ATmega2560	ATmega328P	Broadcom BCM2835	Broadcom BCM2837
Operating Voltage	5V	5V	5 V	5 V
Input Voltage (recommended)	7-12V	7-12V	5 V	5 V
Input Voltage (limit)	6-20V DC	6-20V DC	5 V DC	5 V
Digital I/O pins	54 (of which 15 provide PWM output)	14 (of which 6 provide PWM output)	40-pin GPIO header	40-pin GPIO header
Analog Inputs Pins	16	6	0	0
Current from 3.3V Pin	50 mA	50 mA	50 mA	50 mA
Flash Memory	256 KM of which 8 KB used by bootloader	256 KM of which 0.5 KB used by bootloader	Up to 64 GB	Up to 64 GB
SRAM	8 KB	2 KB	512MB of LPDDR2 SDRAM	1 GB RAM
Clock Speed	16 MHz	16 MHz	1 GHz	1,2 GHz
Price	€44	€19	€5	€35

The two Raspberry microcontrollers have many benefits, such as big SRAM and Flash memories. They also have faster clock speed than the Arduino, but they aren't using microcontroller team is trained for. They are also using different instruction architectures.

Raspberry Pi boasts in high-end specifications but does not provide analogue to digital converter in it's hardware. For our project we need enough digital pins to attach communications module and at least one analogue input.

3.2 Consensus

All platforms could perform required tasks of the project fairly easily but for raspberries we would need to find a way to convert analogue signal to digital. Also Raspberries working on ARM processors for which our team will not receive appropriate training.

For Arduinos Uno would be cheaper and help would be more readily available since it's more popular platform. Arduino Mega2560 has slightly higher specs and most importantly uses the microcontroller we receive training for.

Since school can provide us with Arduino Mega2560 and thus cost not being an issue our team is currently opting for Arduino Mega2560 as our platform.

4. Sensor selection

4.1 Context of use and constrains

The purpose of the sensor in this project is to know whether the trash to be emptied or not. With this in mind, we'll look for a sensor that can help us differentiate between a (nearly) filled bin, and one that is not. We don't want the bin to become too full, nor do we want the sensor to make us believe that the trash is full. The latter may occur if the sensor's view is obstructed, for example.

The sensor should be able to help us differentiate between <90% and >90% filled bins.

In other words, we are looking for a threshold. We need to know when the trash is about 90% filled instead of 100%, as we don't want the trash to be excessively full.

We are most likely going to take a sensor to transmit this binary information (need emptying or not), but the way we use the sensor or the type of information might ultimately change with what sensors we find.

Ideally, the sensor also has to be sturdy and to be able to resist the elements, since it will be placed inside the trash container.

4.2 Characteristics constraints

Range and span: About 0 to 1 meters (I.e. the bin depth's) if we put the sensor on the top of the bin, looking down.

Linearity: Either linear or not should be fine. The optimal one might be a logarithmic scale, since information we need is a threshold.

Sensitivity: Depends on linearity. The choice will be made in the light of the sensor's sensitivity around the threshold's measurement area.

Resolution: Around 5cm (less than 5% of the bin's height), only important when measuring around the threshold area.

Accuracy: Should be less than 10% of the bin's height.

Response time: We can afford a long response time, since garbage retrieval is on a daily magnitude.

Zero: The zero will be set up for an empty trash.

NB: descriptions about bin height are valid only if we opt for a metric measurement.

4.3 Assess options

For the design of our prototype, you can use different types of proximity sensors. The most common are the position switches, capacitive sensors, inductive and photoelectric, infrared and ultrasonic sensors.

Here are the main features of each type of sensors and compares to justify the choice of the final sensor.

4.3.1 Position switches

Position switches or limit switches, are a type of sensors that act when it detects direct contact. Normally placed at the end of the route of a machine and are activated when they perceive contact. Such sensors are invalid for our project because we do not want contact.

4.3.2 Capacitive sensors

Capacitive sensors are suitable for the detection of metallic and non-metallic objects of all types. Even highly transparent glasses or liquids are easy to detect with a capacitive sensor. Only the dielectric conductivity of the target material is relevant. The greater the dielectric constant of a material, the higher the possible switching distances or the more reliable detection.

We decided to looking for capacitive sensors which have a thread size of 30 because it’s where we find the best switching distance. In table 3 we are comparing different capacitive sensors.

Table 2 Capacitive sensors

Name	Switching distance	Installation	Adjustment	Switching output	Connection type
KD/KL 30	1 - 20 mm	Flush / non-flush	Potentiometer	PNP or NPN	Plug M12 4-pin
CM30-25NNP-EW1	4 - 25 mm	Non flush	Potentiometer	NPN or PNP	Cable, 4-wire, 2m (do not bend below 0 °C)
CM30-25NNP-EW1	4 - 25 mm	Non flush	Potentiometer	NPN or PNP	Male connector M12, 4-pin
CM30-16BPP-KC1	2 - 16 mm	Flush	Potentiometer	NPN or PNP	Male connector M12, 4-pin
CM30-16BPP-KC1	2 - 16 mm	Flush	Potentiometer	NPN or PNP	Cable, 4-wire, 2m (do not bend below 0 °C)

The disadvantage of this type of sensor is the scope, because depending on the diameter of the sensor, can reach up to 60 mm. Another drawback is that depends on the mass to be detected, if you want to perform a detection of any type of object this sensor does not work, since it depends on the constant electric.

There are other drawbacks, but we have basically ruled out this option for the little measurable length. In the end we chose not to opt for capacitive sensors.

4.3.3 Inductive sensors

Inductive sensors are a special kind of sensors used to detect ferrous materials. They are of great use in industry, both for positioning applications to detect the presence or absence of metallic objects in a particular context. We have also ruled out such sensors because they are not what we want for our application. Especially because there is different type of materials in the trash, not only metal.

4.3.4 Photoelectric sensors

A photoelectric sensor or photocell is an electronic device that responds to the change in light intensity. These sensors require an emitter that generates light component and a receiver component that receives the light generated by the issuer.

Again we discarded this type of sensors because inside the garbage containers, the light is low and therefore the measures would not be accurate. Also they are not used for long distances.

4.3.5 Infrared sensors

Infrared sensors operate by emitting a signal of infrared light, not perceptible by the human eye, which is reflected in the material, and according to the time it takes to get the reflected signal, can measure the distance. This type of sensor is very useful for short distances, but not for long distances as we need for our prototype.

4.3.6 Ultrasonic sensors

Ultrasonic sensors are proximity sensors working free of mechanical friction and detect objects at distances up to 8m. The sensor emits ultrasonic pulses, then they reflect an object, the sensor receives the echo produced and converted into electrical signals, which are produced in the apparatus of valuation. These sensors work only in the air, and can detect objects with different shapes, surfaces and different materials. The materials can be solid, liquid or dusty, however they must be sound baffles. The sensors measure the time course of the echo, that means the time between the transmitted pulse and echo pulse is evaluated.

Table 3 Ultrasonic sensors

Name	SEN136B5B	HC-SR04	SRF05	MB1202
Price (Kr)	89.55	14.13	14	Embedded Stock
Max Range (cm)	600	400	400	765
Min Range (cm)	3	2	1	25
Power Supply (V)	5	5	5	3-5
Resolution (cm)	1	0.3	1	1
Effectual Angle	<15°	<15°	<15°	N/A
Measuring Angle	30°	30°	30°	N/A
Dimension (mm)	43x20x15 (mm)	43x20x15 (mm)	43x20x17 (mm)	22x20x25 (mm)
Pin number	3	4	5	5

4.4 Consensus

All of the sensors above fill requirements for our application. School's embedded stock has model MB1202 I2CXL-MaxSonar-EZ0 available for rental. It also has highest range and large enough beam for our application. Our project doesn't need to use sensor in an angle. MB1202's availability and attributes make it supreme to other options.

5. Communication module selection

In our project communications plays a critical role. Our goal is to accomplish Internet of things device out of garbage can. Our communication technology should pass following requirements:

- Work over long distances
- Be able to transmit data in TCP/UDP packages
- Be compatible with our platform
- Have good documentation
- Cost under 100 euros
- Work in variable conditions (cold, rain etc.)

5.1 Assessment of options

To choose best technology for the situation analysis was made out of different kinds of communication technologies.

5.1.1 Wi-Fi

Wireless networking had supreme documentation and modules available easily. It also supports very high data-transfer rates compared to some of other technologies available. Problems with Wi-Fi are packet loss due collision or bit errors. Also for our application distances would be too long to cover with Wi-Fi network.

5.1.2 Bluetooth

Bluetooth provides easy machine to machine connections with good documentation. Modules are cheap and communication with smartphone would've been easy. For our application Bluetooth's range is unusable and thus had to be discarded as an option.

5.1.3 LoRA

New technology being developed offers data transmission at low bitrate. Maximum package size for LoRa is 250 bytes. Range for LoRa is documented at 21.6 km which could be enough for our project with good placement of receiver. Problems with LoRa is inability to do M2M connections and poor documentation on development. Modules with receiver are also expensive.

5.1.4 GPRS

GPRS is packet switching technology that works on GSM network. It allows TCP and UDP connections and covers whole city. Modules are available at school's embedded stock. Documentation and examples found are good. Data rates for GPRS are low and it doesn't offer quality of service techniques. Uses AT commands to operate.

5.1.5 3G

Newer technology alternative to GPRS that works over mobile network. Provides fast mobile data with TCP and UDP connections. Also uses AT commands to operate. Documentation on development is hard to find. Modules also tend to be more expensive.

5.2 Consensus

Between available technologies three best for our application are LoRa, GPRS and 3G. Due to LoRa being more expensive and poorly documented, communications technology used for our project will be either GPRS or 3G.

In table 4 we compare properties of five communications modules.

Table 4 Communications module comparison

Name	ADH8066 GSM Module	Cellular Shield SM5100B	Arduino GSM Shield 2	Cooking Hacks SIM5215E	Cooking Hacks SIM908
Frequency	GSM 50/900/1800/ 1900	GSM 850/900/18 00/ 1900	GSM 850/900/180 0/ 1900	GSM 850/900/ 1800/1900 WCDMA 2100M/1900M/850M.	GSM 850/900/1 800/ 1900
Working temp	-30°C +70°C	-10°C +55°C	N/A	N/A	-40°C +85°C
Communications	GPRS	GPRS	GPRS	3G	GPRS
Antenna	External	External	External	External	Internal
GPS	No	No	No	No	Yes
RTC	No	Yes	No	No	No

Price	Embedded stock	Embedded stock	€71.50	€149	€99
Extras			GSM-library support		

For our project, best choice would be SIM908 since it has antennas + GPS built in.

6. Server selection

6.1 Assessment of options

We will discuss about different free servers, theirs features, how they work.

6.1.1 Ubidots

Ubidots is an Internet of Things and works with Arduino, Raspberry Pi. We can find tutorials to learn how to use it with Arduino. It uses an ID for each data and a API key to identify where to store values. Nevertheless, it will be difficult to parse data into our app because we don't study this.

6.1.2 ThingSpeak

ThingSpeak is an open source Internet of Things application. It allows us to analyse and visualize data using MatLab. We can find a lot of features but useless for us given that we only need to store two values.

6.1.3 Xively

Xively is also is a secure platform for Internet of Things devices and products. Its API and web service provide real-time control and data storage. To store data, it works like Ubidots except that instead of use an ID, we use the name of the channel. We can find tutorials also on the web. To parse data, we can use JSON and we already know how it works as we studied it in ITSMAP courses.

6.2 Consensus

We could use all of them to our project but to save time it will be easier to use Xively because we already know how to parse it.