Switchack

TeamCode

Faculty of Engineering - Helwan University

2017-2018

Document Change Log

Version	*Date*	*Author*	*Changes*
0.1	*25/9/2016*	*MIE Committee*	*Basic draft for project overview.*
1.0	*30/12/2016*	*MIE Committee*	*Final MSC changes – adding MIE 2017 Tracks*
1.1	*31/10/2017*	*Switchack team*	*Project overview*
/			

Project Details

Problem definition

Egyptians across the country are becoming increasingly frustrated with the gross increase of electricity bills in recent months, Where prices increased by an average of 33% for all segments. This problem face almost all the Egyptians and they can do nothing about this increase but paying bills to use the electric power. And from here our idea came, We intend to minimize electricity usage by knowing electricity usage itself which will help individuals saving the wasted or the unwanted electric power in their everyday.

Problem Solution

We propose to solve this critical problem using a user friendly mobile application and a smart Electrometer could be installed at your home besides your electrical panel to measure your usage and give you an estimation on how much your next bill is likely to be, recommend how to economize your usage and you can also determine the power of your device and the usage you do not want to exceed, so the application will recommend the best way to limit it.

MIE tracks:

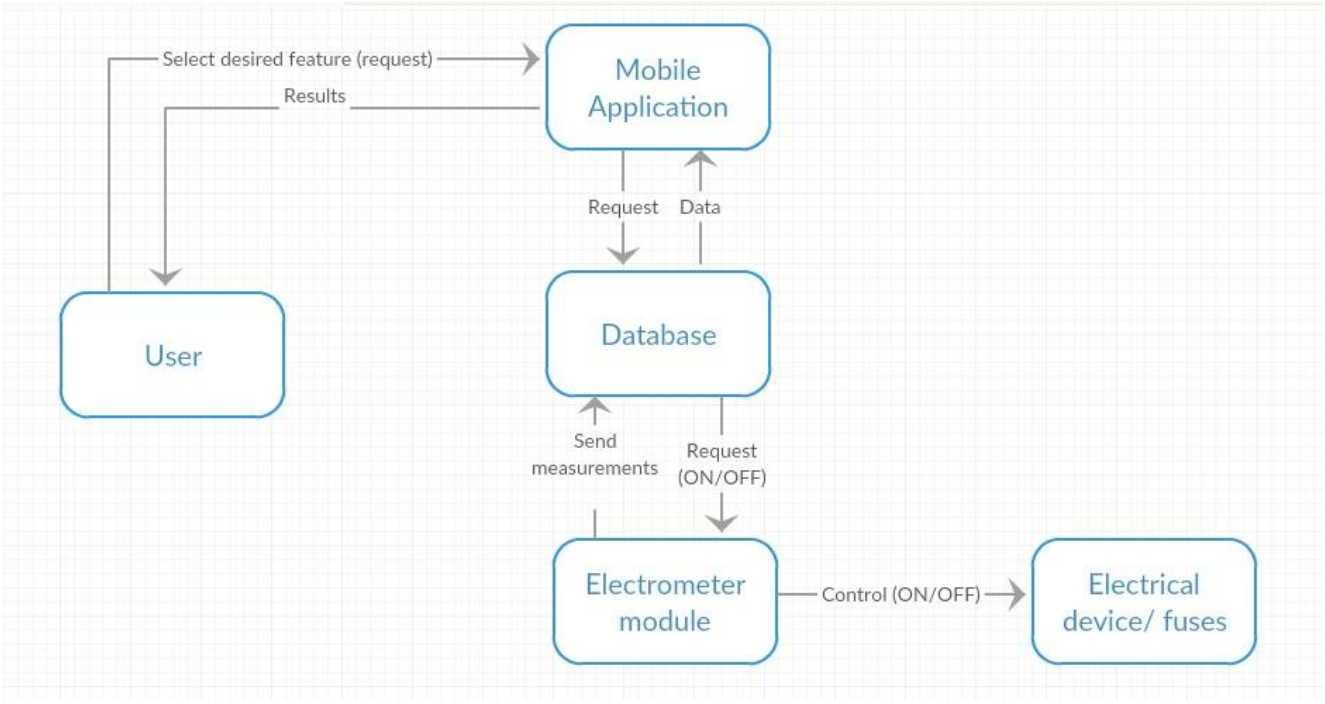
Project Track: *Internet of Things (IOT)*

Technical Approach

1- Project's Output description: Feature List

	Feature Name	Brief Description
1	*Maintainability*	*We can perform a successful repair action within a given time. so system can be restored to operational status after a failure occurs.*
2	*Scalability*	*The device can be used in different places with different number of fuses*
3	*Usability*	*We develop a user friendly mobile application to make it easy to contact with the device*
4	*Security*	*We use Firebase database platform to make our users and their data safe & secure*
5	*Remote Control*	*Users can remotely turn ON/OFF any fuse.*
6	*Applicability*	*Project is applicable to all types of homes, smart home and not smart homes.*

2- Block Diagrams



Switchack context diagram

3- Software/ Hardware Development Tools and Components

Component	Usage/justification	Price	Store / Datasheet
Raspberry Pi	Control all the sensors and relays	1400	RAM Electronics
Current Sensor	Measure current to get usage information	4 X 75	RAM Electronics
Relays	Switching on/off the circuit breaker in electric panel	100	RAM Electronics
Test Board	Testing the project's components	300	RAM Electronics
Electric Board	Getting 220 volts & connecting the devices	200	RAM Electronics
12-Channel On-Board AC Current Monitor for I2C	Current monitoring of the AC power loads using cross-platform I2C communications, making current monitoring possible for your Raspberry Pi, BeagleBone, Arduino, or any other computing platform that supports I2C	2400	Control everything online store

	but we will use it as a plan B.		
Android Studio	*The official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA .*		
Java	*the used programming language for android apps implementation*		
XCode9	*The Integrated Development Environment (IDE) for IOS app development*		
Swift	*the used programming language for ios apps implementation*		
Python	*the used programming language for Raspberry pi*		
IDLE (Integrated DeveLopment Environment or Integrated Development and Learning Environment)	*is an integrated development environment for Python,*		
Firebase	*Database platform*		

4- Design and Implementation

Switchack is a hardware module that is made with iot technology to make a connection between users and the modules through their mobile phone and to make our project achievable there are some obligatory requirements and knowledge we must have/gain:

- *Android Development*
- *Internet of things (IOT)*
- *Firebase*
- *IOS Development*
- *Data Analysis*
- *Python*
- *Embedded Systems*

Switchack is divided into 4 modules

1. ***Software Module:***

We aim to release a user-friend ease running application for both android and ios to make sure that all individuals that want to control their electricity usage all are well served.

2. ***Hardware Module:***

We aim to achieve reliability to our users so we went with RaspberryPi kit with some sensors and relays and that will help us making a good hardware which is easy in installation and setup.

3. ***Communication:***

This module is from the most important parts in our project as it links between the hardware module and the software referring to the database platform we use (Firebase).

4. ***Installation:***

Here we assure that our product needs only a technician to be installed to your electrical panel even though you don't have a smart home.
All you need the hardware installation and the software application installation from Apple store and Google store.

Project start point:

After dividing the project to four modules, we also divide the team into three working groups:
1. *Hardware team*
2. *IOS development team*
3. *Android development team*

Ø Expected integration effort :
Each one of the modules mentioned earlier will be implemented and tested separately as a black box then the integration phase will take place. Integration between these modules will be through the database servers.

Ø Expected challenges:
1. *The electric panels in Egyptian homes are not similar to each other.*
2. *Reading data from the current sensors & pushing data to used servers instantly and accurate.*
3. *Security of the hardware module and the software applications.*

The current implementation status:

We bought almost all the needed components required to send and receive the data of usage and we started designing software applications' user interface.

The progress status in other parts:

Hardware already sends required data and now it is tested to check the accuracy and there is 2 designing layers are implemented and it's being tested now.

5- Master Project Time Plan

All the phases is in parallel with documentation of each phase (including marketing and business studies)

Phase 1: Team Formation

We started seeking people that are qualified enough to work in IOT track and passionate enough to go for miles and we also searched for advisors to support the project with a good background in iot.

Phase 2: Ideas Brainstorming & selection

We searched for critical problems faced by Egyptians in the last few months which is the Electricity is the most common problem and how it can be solved by IOT

Phase 3: idea surveying:

After surveying Electricity problems faced by people, we found the most common problem is the high cost bills and we started to ask them the best way to figure out that problem and using which technologies and the most easily devices can be used through our project

Phase 4: Collecting Requirements:
We started to search about the knowledge that we should have/gain in the next few months to start the implementation phase to release the desired prototype.

Phase 5: Gaining knowledge:
Through some online courses.

Phase 6: implementation:
Due to our structure we are using modules and we have a team for each module

Phase 7: unit testing

Each individual unit/component is tested after implementation

Phase 8: software and hardware Integration
Linking the software and hardware by database servers

Phase 9: integration testing
Individual modules are combined and tested as a group.

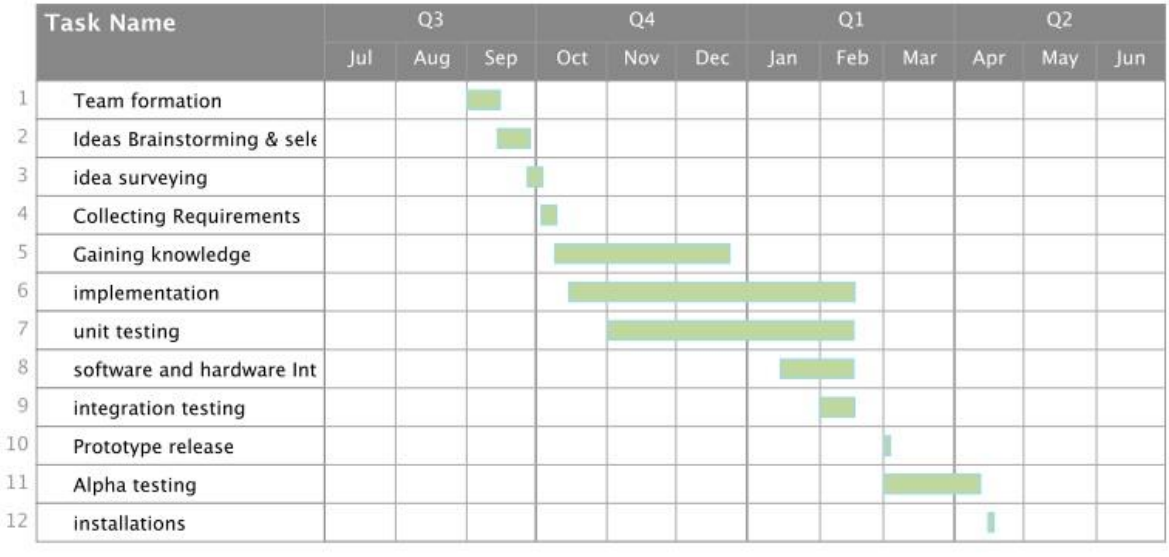
Phase 10: Prototype release

Phase 10: Alpha testing
The focus of this testing is to simulate real users by using blackbox and whitebox techniques. The aim is to carry out the tasks that a typical user might perform.

Phase 12: installations
The application will be available on play store and apple store, and the hardware device will be installed on the user's electrical panels.

Task Name	Duration	Start	Finish
Team formation	2w	09/01/17	09/14/17
Ideas Brainstorming & selection	2w	09/14/17	09/27/17
idea surveying	1w	09/27/17	10/03/17
Collecting Requirements	1w	10/03/17	10/09/17
Gaining knowledge	2 months	10/09/17	12/23/17
implementation	4 months	10/15/17	02/15/18
unit testing	77d	11/01/17	02/15/18
software and hardware Integration	24d	01/15/18	02/15/18
integration testing	11d	02/01/18	02/15/18
Prototype release	1d	03/01/18	03/01/18
Alpha testing	30d	03/01/18	04/11/18
installations	1d	04/15/18	04/15/18

Switchack



Business Model (Market and Competition)

`

1- Market analysis

- *Marketing segmentation:*

*By using **Demographic segmentation** as a one of the simplest and most widest type of market segmentation used.*

#	Market Segment	Product features
1.	*Income: Moderate*	- *Moderate price to be suitable for anyone who want to consume their usage.*
2.	*Age: between 21-45*	- *User friendly mobile application*
3.	*Marital status: Families*	- *Secure home energy monitoring system that helps you take control of your house.*
4.	*Capable people*	- *control of home electric energy through their mobile phone*

After surveying and collecting opinions of some people who are interested in the project idea:

#	Customer need	Metrics	Priority
1.	*Fast Response*		*High*
2.	*Ease of use*		*High*
3.	*suitable price*		*High*
4.	*low maintenance cost*		*Moderate*
5.	*Security*		*Moderate*
6.	*Safe to operate*		*High*

2- Competitor analysis

#	Competitors	Capabilities	Weakness	Competitive Edge
1	*Energy Consumption Analyzer*	*This application keeps track of your energy consumption. Add your meters for gas, electricity, or water to the database and record the current meter readings from time to time. Readings can be color-coded and comments may be added to remember special situations which may*	*. Users have to add their measurements manually which waste time and effort*	- *The hardware device send all measurements to the mobile application automatically, so it's easier and more accurate* - *Users can also determine the power of their*

		explain unusual energy usage		*device and the usage they do not want to exceed, so the application will recommend the best way to limit it.*
2	*Smappee*	*With the Smappee energy monitor you always know how much power your appliances consume, wherever you are. For more comfort, a better insight and sustainable energy savings up to 30%.*	*. Users can not control their usage remotely. It's just an energy monitor*	- *With Switchack, you can enable or disable any fuse through the mobile application*
3				

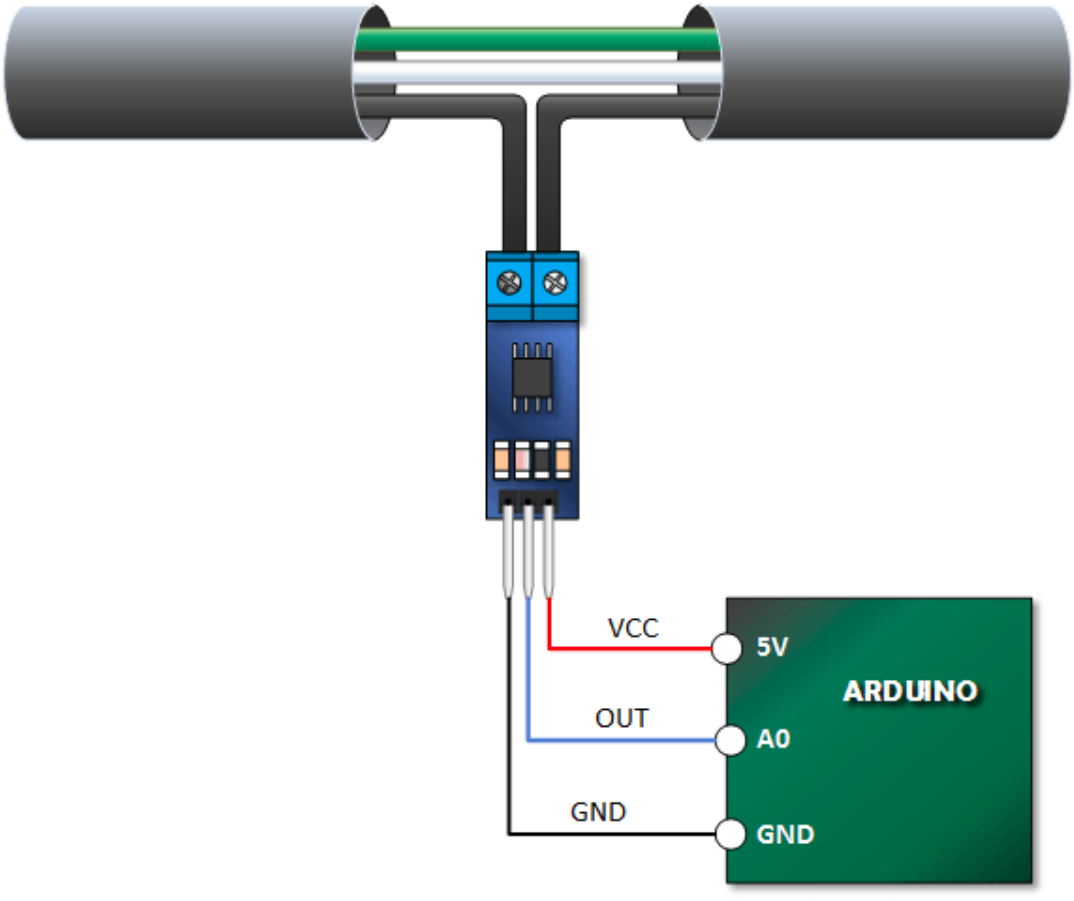
3- Financial Analysis

1- Development cost
Total components Cost: 5000 EGP (including spare parts)
Expected development cost: (5000 + 1000) = 6000 EGP

2- Fund raising:
At this moment we don't have any sponsor for technical or financial support, but we are planning for getting sponsors and participating in some competitions to find the technical, marketing and financial support.

Project Photos/Screenshots

Connect the components as shown below:



```
import firebase_admin
from firebase_admin import credentials

# Import database module.
from firebase_admin import db

# Fetch the service account key JSON file contents
cred = credentials.Certificate('graduation-4da3f-firebase-adminsdk-j4x7f-cc56f76

# Initialize the app with a service account, granting admin privileges
firebase_admin.initialize_app(cred, {
          'databaseURL': 'https://graduation-4da3f.firebaseio.com'
        })
x=7
while(1):
        # Get a database reference .
        ref = db.reference('users')

        ref.child("0102").set({

                'value': x,


        })


        # Read the data from firebase (this is a blocking operation)
        print(ref.child("0102").child("value").get())
```

.

```
#!/usr/bin/env python
#
# Analog Input with ADC0832 chip
#
# Datasheet: http://www.ti.com/lit/ds/symlink/adc0838-n.pdf
# Part of SunFounder LCD StarterKit
# http://www.sunfounder.com/index.php?c=show&id=21&model=LCD%20Starter%20Kit
#

import time
import os
import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BCM)

# change these as desired - they're the pins connected from the
# SPI port on the ADC to the Cobbler
PIN_CLK = 18
PIN_DO  = 27
PIN_DI  = 22
PIN_CS  = 17

# set up the SPI interface pins
GPIO.setup(PIN_DI,  GPIO.OUT)
GPIO.setup(PIN_DO,  GPIO.IN)
GPIO.setup(PIN_CLK, GPIO.OUT)
GPIO.setup(PIN_CS,  GPIO.OUT)

# read SPI data from ADC8032
def getADC(channel):
        # 1. CS LOW.
        GPIO.output(PIN_CS, True)       # clear last transmission
        GPIO.output(PIN_CS, False)      # bring CS low

        # 2. Start clock
        GPIO.output(PIN_CLK, False)  # start clock low
```

```
        # 3. Input MUX address
        for i in [1,1,channel]: # start bit + mux assignment
                if (i == 1):
                        GPIO.output(PIN_DI, True)
                else:
                        GPIO.output(PIN_DI, False)

                GPIO.output(PIN_CLK, True)
                GPIO.output(PIN_CLK, False)

        # 4. read 8 ADC bits
        ad = 0
        for i in range(8):
                GPIO.output(PIN_CLK, True)
                GPIO.output(PIN_CLK, False)
                ad <<= 1 # shift bit
                if (GPIO.input(PIN_DO)):
                        ad |= 0x1 # set first bit

        # 5. reset
        GPIO.output(PIN_CS, True)

        return ad

if __name__ == "__main__":
        while True:
                print "ADC[0]: {}\t ADC[1]: {}".format(getADC(0), getADC(1))
                time.sleep(1)
```

Appendix A

`

END OF REPORT