# Brief

This EMS firmware utilize mbed timer interrupt. These are the major functions that construct EMS firmware.

# Major Functions

LCD Display

|  |
| --- |
| Update Display |
| Update available interface |
| Update energy consumption |
| Update switch state |
| Update IR remote and temperature |
|  |
| Period: 5 sec |

Connectivity

|  |  |
| --- | --- |
| Check Connectivity | Connect to Network |
| Check ETH link | Connect to ETH |
| Check WiFi link | Connect to WiFi (optional) |
| Check GPRS link | Connect to GPRS (optional) |
|  |  |
| Period: 30 sec | Period: 5 sec |

Energy Measurement

|  |
| --- |
| Measure Energy |
| Measure energy |
| Backup data to SD card |
|  |
| Period: 1 sec |

Switch Control

|  |  |
| --- | --- |
| Get Control Command | Control Switch |
| Get control command from server | Read command from SD card |
| Save command to SD card | Execute command |
|  |  |
| Period: 10 sec | Period: 10 sec |

# Procedural Firmware

The firmware consist of multiple modes:

1. Registration
2. Operational
3. WiFi Setting
4. Remote Discovery
5. Remote Setting

Global initialization:

1. Init peripheral and iface (e.g. TFT)

## Registration Mode

Single loop:

1. Update LCD display
2. Init panel parameter (e.g. SSID, PASS)
3. Connect to network
4. Check connection
5. Get panel parameter (based on WiFi MAC address) from cloud server
   * getCmdURL, getCmdKey, cmdExecURL, cmdExecKey, sendDataURL, sendDataKey
6. Update LCD display with getCmdKey
7. Verify registration with getCmdKey
8. Save panel parameter

Notes:

1. Get panel parameter from cloud server
   * Cmd:

**192.168.128.69/emma/data/register18fe349936be.json** (18fe349936be is WiFi MAC address)

* + Response:

**[{"getCmdURL":"(/talkbacks/887/commands/last?api\_key=)","getCmdKey":"(ACM8XW24UDVY1GTV)","cmdExecURL":"(/execute?api\_key=)","cmdExecKey":"(2)","sendDataURL":"(/update?api\_key=)","sendDataKey":"(99N5T6AH3NY4UKNA)"}]**

1. Verify registration with getCmdKey
   * Cmd:

**192.168.128.69/emma/data/verifyACM8XW24UDVY1GTV.json** (ACM8XW24UDVY1GTV is getCmdKey)

* + Response:
    1. Success: **[{"status":"registered"}]**
    2. Failed: **[{"status":"failed"}]**

## Operational Mode

Initialization:

1. List of Remotes ✓
2. Init panel parameter (e.g. SSID, PASS) ✓
3. Connect to network ✓
4. Check connection ✓

Loop:

1. Update LCD display
   1. Interface ✓
   2. Energy data 🗶
2. Connect to network ✓
3. Check connection ✓
4. Get command from server ✓
5. Execute command
   1. Parse command ✓
   2. Exec command
      1. Remote ✓
      2. Switch 🗶
   3. Send execution status 🗶
6. Send energy data to server 🗶

Notes:

1. Command types:
   * Remote Control
     + Example: **[{“type”:0},{"id":”234”,"mac":"00262903424E","cmd":1},{"id":”345”,"mac":"00262903424F","cmd":2}]**
     + Means run CMD1 file to device with mac = 00262903424E…
   * Switch Control
     + Example:

**[{“type”:1},{"id":”234”,"sw":”1”,"cmd":1},{"id":345,"sw":”2”,"cmd":0}]**

* + - Means switch sw 1 to 1 (ON) and sw 2 to 0 (OFF)
  + Notes
    - Command service should be consumed only once

1. Energy data
   * Example:

**http://api.thingspeak.com/update?api\_key=99N5T6AH3NY4UKNA&field1=90**

## WiFi Setting

Initialization:

1. Init panel parameter
2. Init WiFi Server

Loop:

1. Update LCD display
2. Receive data
3. Check data
4. Save SSID and PASS data

## Remote Discovery

Single loop:

1. Update LCD display (this mode only support ETH)
2. Discover remote
3. Update LCD (MAC – IP – Label)
4. Save remote list