

## System construction:

The system is a smart controlled house and is consisted of 4 Electronic control units (ECUs). The names and responsibilities of each ECU is summarized below:

- 1- Remote control ECU (RC ECU):
  - It is a hand held device.
  - Provides human interface for access and security control.
  - Wireless interface with the main ECU through Bluetooth interface.
- 2- Main Unit and Gateway ECU (MUGW ECU):
  - Wireless interface with the Remote Control ECU through Bluetooth.
  - Security verification after request from the RC ECU.
  - Passing different commands from the RC ECU to the SR and AD ECUs.
  - Containing latest configurations requested by the EC ECU.
  - Controlling digital calendar calculation and display.
- 3- Sensor Reading ECU (SR ECU):
  - Interfacing 3 LM35 temperature sensors.
  - Interfacing 5 motion detection sensors.
  - Interfacing 3 light sensors.
  - Provide commands to the AD ECU based on the sensor data and the received configuration from the MUGE ECU.
- 4- Actuator drive ECU (AD ECU):
  - Driving 5 RGB LEDs.
  - Driving 3 LEDs simulating Air Conditioner motors.
  - Driving 5 LEDs Simulation Doors.
  - Driving 3 LEDs simulation light control.

## Interface framing:

The interface between the RC ECU and the MUGW ECU is done through Bluetooth. You can use a Bluetooth module with the part number HC-05. The exchanged frames between the two ECUs is consisted of the following format.

Frame Header	Frame type	Data Size	Data Bytes			Checksum	frame tail
0xaa	0..9	0..6	Xx	xx	Xx	Xx	0x55

- **Frame header:**  
Is used to indicate the frame starting to allow the receiver to know when to start parsing.
- **Frame type and data bytes:**  
Is used to inform the receiver with the type of request. Frame types are summarized in the following table.

Frame Type ID	Frame meaning	Notes
0x00	Self test	No data will be transmitted in this command
0x01	check the contained pass key	
0x02	pass key status	in the data, 0x00 means OK and 0x01 means not OK
0x03	Set the contained passkey	

0x04	Get number of wrong pass key tiers	
0x05	set calendar with the contained data	seconds, minutes, hours, day, month and then two bytes for year
0x06	get calendar data	seconds, minutes, hours, day, month and then two bytes for year
0x07	Set room n with the contained temperature threshold	3 bytes, 1st is room ID, 2nd is the integer part of temperature and third is the fractional part with fixed point representation

0x08	get room n temperature threshold	3 bytes, 1st is room ID, 2nd is the integer part of temperature and third is the fractional part with fixed point representation
0x09	Request testing door open	1 byte is the door id
0x0a	Request testing door closed	1 byte is the door id
0x0b	Set Room n RGB Light	4 bytes. 1st is the room id. Second is the Red , 3rd the green , 4th is the blue

0x0c	Get Room n current RGB Light	4 bytes. 1st is the room id. Second is the Red , 3rd the green , 4th is the blue
------	------------------------------------	---

- **Checksum:**  
Is the xor result of all the bytes from the frame header till the last data bytes. Request will not be performed if checksum was wrong.
- **Frame tail:**  
Used to indicate that the frame data ended.

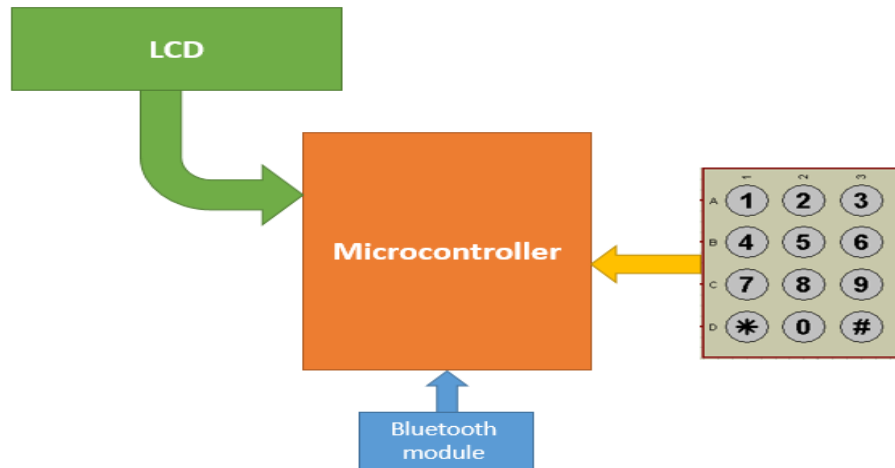
The interface between the MUGW, AD and SR ECUs is done through CAN protocol or I2C protocol. Any ECU can be master which only works on the master transmit mode and if it was a slave, it will work on the slave receive mode. The frame IDs and data are the same as the predefined Bluetooth module. The Slave addresses for each ECU is summarized below.

ECU	Slave Address
MUGW	0x79
AD	0x7A
RS	0x7B

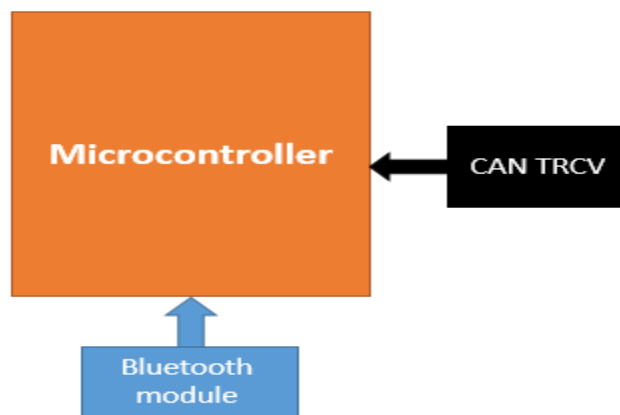
If the communication is done through CAN, the message IDs from Each ECU will be its predefined slave address, and the data bytes is identical to the Bluetooth frame.

## ECUs Construction:

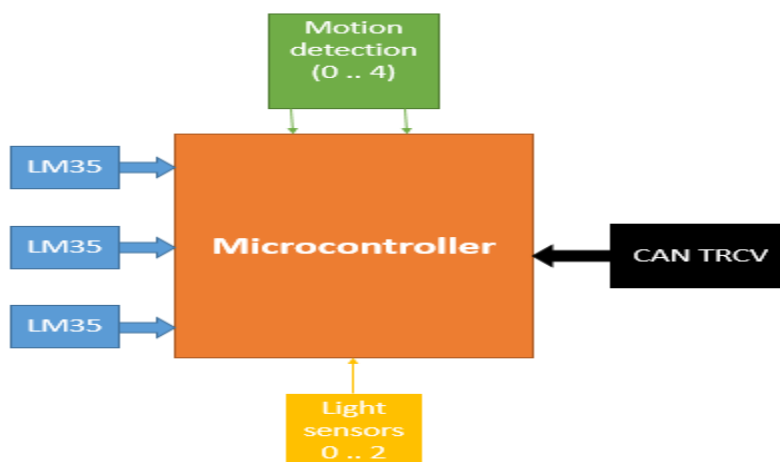
- RC ECU:



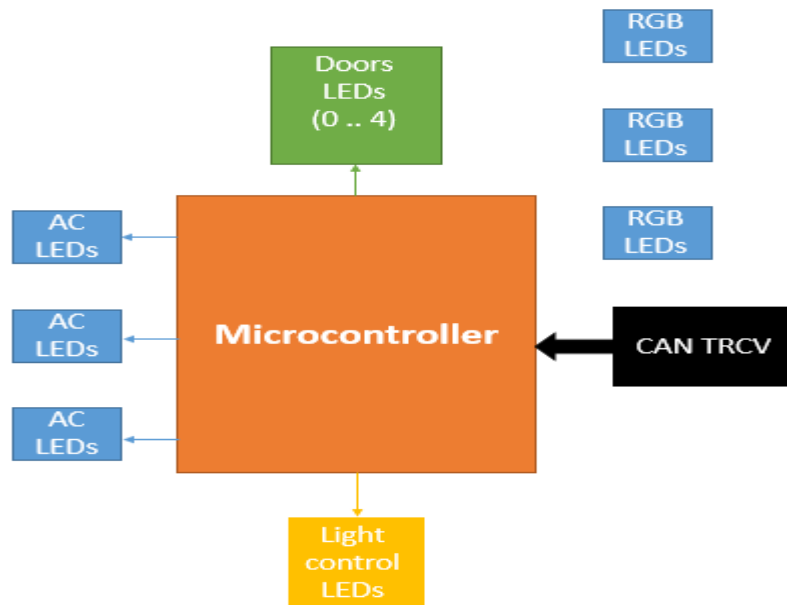
- MUGW ECU:



- RS ECU:



- AD ECU:



## System operation:

### 1- Self test:

After power up, the RC ECU will first send a **self test** frame to the MUGW ECU. The MUGW ECU will then send a **self test** frame to the AD and SR ECUs. If the AD and SR ECUs are powered up they will respond with a **self test** frame on the CAN/I2C bus. That means the MUGW ECU will be informed that the other ECUs are working by receiving two **self test** frames from the I2C/CAN bus. If that happened, the MUGW ECU will respond to the RC ECU with a **self test** frame. Finally, if the RC ECU received a **self test** frame within 2 seconds, it will proceed to the next step else it will display an error message and then switch off.

### 2- Security access:

After self test, the RC ECU will display a welcome message to the user and then ask him to enter the passkey. The passkey is consisted of 6 digits, all displayed on the screen as '\*'. The used keypad is a 3x4 phone keypad. The number buttons are used to enter passkey as one number per every press, '\*' key is used to delete one digit and the '#' key is used to confirm the entered passkey and The '#' key be active only when 6 digits are entered.

After entering passkey, the RC ECU will send a frame with the ID 0x01 containing the entered passkey to the MUGW ECU. The MUGW ECU will compare the received passkey with the one saved on its EEPROM. (default one is 000000). The MUGW ECU will respond with a frame with ID 0x02, containing the passkey status and the number of available wrong triers (3 after right pass and is being decremented after each wrong trier).

In the RC ECU and after receiving the Passkey status, it will check it and request the user to enter password if check wasn't ok and number of triers is higher than 0. It will proceed to the next step if the passkey status was ok. It will halt the system for 5 seconds if the number of triers was 0 and then repeat the operation again.

### **3- Options screen:**

After security access succeeded, the RC ECU will display some options to the user to select. The user will move between options using 2 and 8 keypad buttons (up and down). It will use the '#' key to confirm option selection. The options are: Change passkey, Digital calendar view and update, A/C temperature control, Self test requests, and RGB light control.

### **4- Changing passkey:**

If the selected option was changing passkey the RC ECU will display a message to the user to request enter the new passkey with the same previous specs. After entering the passkey by the user, the RC ECU will send a frame with the ID 0x03 containing the new passkey. The MUGW ECU will receive the frame, extract the passkey and save it on its internal EEPROM.

### **5- Digital calendar:**

If the selected option was digital calendar, the RC ECU will send a periodic frame with the ID 0x06 to get the calendar data. The MUGW ECU will respond to this frame with the frame ID 0x05 that contain the current calendar data. The periodicity of the frame and response is 100 ms. From the power up, one of the MUGW ECU tasks is to use the timer RTC to count the number of passed seconds and updates its calendar data. When the RC ECU send the Frame with ID 0x06, it requests the current calendar data. The RC ECU is continuously display the received calendar data to the user with the format "dd:mm:yyyy" on the first line, and "hh:mm:ss" on the second line. If the user pressed the key '#' the RC ECU will start changing calendar operation. If the user pressed the key '\*', the RC ECU will cancel the current operation and back to the options screen.

If changing the calendar requested, the RC ECU will blink each element of the calendar data. The blinked element is meaning that the user can increment or decrement this element. The user will use the keys '2' and '8' to change the element, use the keys '4' and '6' to change the blinked element, use '#' key to confirm the new calendar data.

When the new calendar data confirmed, the RC ECU will send a frame with the ID 0x05 that containing the new calendar data. When the MUGW ECU receive the calendar data it will save it on its memory and continue counting starting from it.

### **6- Self test requests:**

If the selected option was self test requests, the RC ECU will display a room number and a request type (default is OPEN). The display will blink each element. User will use keys '2' and '8' to change the element value, use keys '4' and '6' to change the blinked element and the key '#' to confirm selection.

After the selection confirmed, the RC RCU will send a frame with the ID 0x09 or 0x0a dependent on the selected request type (OPEN or CLOSE). After receiving the frame, the MUGW ECU will send the request as an I2C / CAN frame to the AD ECU which will turn on or off the Doors simulation LEDs.

### **7- RGB control:**

If the selected option was RGB light control, the RC ECU will send frame with the ID 0x0C to get the latest saved RGB selections. The MUGW will send the same request to the AD ECU which will respond a frame



with ID 0x0b after reading the latest data from its internal EEPROM. The MUGW ECU will resend this frame to the RC ECU through Bluetooth.

The RC ECU will display the received data on the screen to the user in terms of four numbers. The first one is the R value, the second one is the G value, the third one is the B value and the fourth one is the RGB LED id. The display will blink each element. User will use keys '2' and '8' to change the element value, use keys '4' and '6' to change the blinked element and the key '#' to confirm selection.

After confirming the selection, the RC ECU will send a frame with the ID 0x0b containing the new RGB data. After receiving the frame, the MUGW ECU will resend it on the I2C/CAN bus to the AD ECU which will save the extracted data on its internal EEPROM.

#### **8- A/C temperature control:**

If the selected option was A/C temperature control, the RC ECU will send frame with the ID 0x08 to get the latest saved temperature selections. The MUGW will send the same request to the SR ECU which will respond a frame with ID 0x07 after reading the latest data from its internal EEPROM. The MUGW ECU will resend this frame to the RC ECU through Bluetooth.

The RC ECU will display the received data on the screen to the user in terms of two numbers. The first one is the temperature value and the second one is the sensor id. The display will blink each element. User will use keys '2' and '8' to change the element value, use keys '4' and '6' to change the blinked element and the key '#' to confirm selection.

After confirming the selection, the RC ECU will send a frame with the ID 0x07 containing the new temperature data. After receiving the frame, the MUGW ECU will resend it on the I2C/CAN bus to the SR ECU which will save the extracted data on its internal EEPROM.

#### **9- SR ECU main operation:**

After power up, the SR ECU will perform a periodic reading to its connected sensors data (temperature sensors, light sensors and motion detection sensor). After that it will compare them with their threshold values on its internal EEPROM (A/C threshold, daylight threshold and motion enter and motion exit). Finally, it will put different frames with the token decision on the I2C/CAN bus to the AD ECU. The periodicity of performing the previous operation is 200 ms.

#### **10- AD ECU main operation:**

After power up, the AD ECU will continuously monitor the CAN/I2C bus to extract the different requests, update its internal EEPROM and finally drive the different actuators (A/C LEDs, Light control simulation LEDs, doors simulation LEDs and the RGB LEDs). The periodicity of performing the previous operation is 100 ms.