**Introduction**

**Hardware platform and devices**

AVR → MIDAS Engineering MDA-Multi MICOM (ATmega8535, http://www.midaseng.com/)

Raspberry Pi 2 (Compatible to RPi3)

Android (Since 4.1 Jelly Bean is available)

Wireless router

USB to RS-232C cable

**Development environment and tools**

**- Compiler** : AVR → IAR Embedded Workbench Atmel AVR C

Raspberry Pi → GCC C Compiler, PHP 5.6.11

**- Operating system** : Raspberry Pi → **Raspbian Jessie 03-18-2016**

Android → Android 4.1 Jellybean or later

**- Development environment :** AVR development → Windows XP or 7 32bit

Raspberry Pi → Ubuntu 15.10 64bit or later

Android → Android Studio 1.5 or later

**- Applications** : Apache2, mariaDB, PHPMyAdmin,

**Installation**

**AVR** → Build CD\_ATmega8535/main.c file and burn into your ATmega8535 or use CD\_ATmega8535/Debug/Exe/CD\_ATmega8535.hex file

**Raspberry Pi 2, 3** → Execute CD\_RaspberryPi2/install.sh in /home/pi/ after expand filesystem and time zone set.(sudo permission and internet connection is needed. Ex) sudo sh install.sh)

**Android** → Build android projects in CD\_Android folder and install in your phone.

**How to use**

Overall structure drawing

Overall system flow chart

**Details of the project**

**AVR ATmega8535**

**DDR pin map**

DDRA → KEY MATRIX = 0x0f

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bit | PA 7 | PA 6 | PA 5 | PA 4 | PA 3 | PA 2 | PA 1 | PA 0 |  |
|  | KEY L3 | KEY L2 | KEY L1 | KEY L0 | KEY C3 | KEY C2 | KEY C1 | KEY C0 | DDRA |
| In/Out | In | In | In | In | Out | Out | Out | Out |  |
| Init Val. |  |  |  |  |  |  |  |  |  |

DDRB → SWITCH, HEX DECIMAL INPUT(Rotary Switch) = 0x00

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bit | PB 7 | PB 6 | PB 5 | PB 4 | PB 3 | PB 2 | PB 1 | PB 0 |  |
|  | IN X3 | IN X2 | IN X1 | IN X0 | HEX A8 | HEX A4 | HEX A2 | HEX A1 | DDRB |
| In/Out | In | In | In | In | In | In | In | In |  |
| Init Val. |  |  |  |  |  |  |  |  |  |

DDRC → LED GREEN, Character LCD DISPLAY(16 x 2) = 0xff

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bit | PC 7 | PC 6 | PC 5 | PC 4 | PC 3 | PC 2 | PC1 | PC0 |  |
|  | LCD E | LCD RW | LCD RS | LED Y0 | LCD D7 | LCD D6 | LCD D5 | LCD D4 | DDRC |
| In/Out | Out | Out | Out | Out | Out | Out | Out | Out |  |
| Init Val. |  |  |  | 1 |  |  |  |  |  |

DDRD → LED RED, UART(RX, TX), STEP MOTOR, SPEAKER = 0xfe

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bit | PD 7 | PD 6 | PD 5 | PD 4 | PD 3 | PD 2 | PD1 | PD0 |  |
|  | STEP | STEP | SPK | LED Y2 | STEP B | STEP A | TXD | RXD | DDRD |
| In/Out | Out | Out | Out | Out | Out | Out | Out | In |  |
| Init Val. |  |  |  | 1 |  |  |  |  |  |

ATmega8535 flow chart

Functions explanations per switch(X0, X1, X2, X3)

**Data transfer protocol information (RPi ↔ AVR)**

※ All command characters are unsigned char.

|  |  |  |
| --- | --- | --- |
| **Command character** | **Command name** | **Available command functions** |
| 0 ~ f  (0x30 ~ 0x39, 0x61 ~ 0x66) | Screen boiler temperature | rs232\_get\_command(‘0 ~ f’)  set\_rs232\_data(‘0 ~ f’) |
| g (0x67) | Loosen gas valve | rs232\_get\_command(‘g’)  stepmotor\_spin(‘g’)  set\_rs232\_data(‘g’) |
| l (0x6c) | Lock door | rs232\_get\_command(‘l’)  door\_lock\_unlock('l')  avr\_sound(‘l’)  set\_rs232\_data(‘l’) |
| p () | 4x4 Keypad is pressed | avr\_sound(‘p’)  set\_rs232\_data(‘p’) |
| u (0x75) | Unlock door | rs232\_get\_command(‘u’)  door\_lock\_unlock('u')  avr\_sound(‘u’)  set\_rs232\_data(‘u’) |
| v (0x76) | Fasten gas valve | rs232\_get\_command(‘v’)  stepmotor\_spin(‘v’)  set\_rs232\_data(‘v’) |
| w () | Process error | set\_rs232\_data(‘w’) |

**ATmega8535 main.c function information (17 functions)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function name** | **Parameters** | **Return datatype** | **Return value** | **Description** |
| delay |  | int | 0 | Delay 65000 counts. |
| doorlock |  | void |  | Show password input screen on LCD. |
| door\_lock\_unlock | u, l | int | 0 | Show password match or stay out mode screen on LCD. |
| stepmotor\_spin | v, g | int | 0 | Rotate step motor left(v) or right(g). |
| init\_rs232 |  | int | 0 | Initialize RS232 registers. |
| set\_rs232\_data | unsigned char | unsigned char | 0 | Send 1byte unsigned char data from AVR via RS-232C cable. |
| get\_rs232\_data |  | unsigned char | 0 |  |
| encryption |  |  |  |  |
| boiler |  |  |  |  |
| SCAN ~ SCAN4 |  | void |  |  |
| init\_devices |  | int | 0 |  |
| avr\_sound | w, u, l, p | int | 0 |  |
| rs232\_get\_command | unsigned char | int | 0 |  |
| encryption |  | void |  |  |
| boiler |  | void |  |  |
| password\_checker |  | int | 0, 1 |  |

**ATmega8535 main.c interrupts information**

|  |  |
| --- | --- |
| **Interrupt name** | **Description** |
| TIMER1\_COMPA\_vect |  |

**Raspberry Pi 2(or 3)**

**avr\_daemon.c**

**control.php**

**Data protocol information**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Command variables** | **Command character** | **Command name** | **Command function** | **\_POST variables** |
| $andClientId | *KEY\_VALUE* | Match ID between android and RPi | strcmp($andClientId, $rpiClientId); | andClientId |
| $andModeGet | 1, 2 | Set command to RPi whether get database or set command to AVR | 1 → andGetMaria(~);  2 → rpiAndCommCheck(~); | andModeSet |
| $andCommGet | 0 ~ f, u,  l, g, v | Request specific function of AVR | avrSetComm($andCommGet, $avrDevId); | andCommSet |

Android (From 4.1 Jelly Bean)