ISSN No. 0976-5697

Volume 8, No. 5, May – June 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Developing Embedded Systems using Embedded C with Atmel Atmega2560 Microcontroller (MCU)

Upendra Singh
Lecturer, Department of CSE&IT
Government Engineering College Bharatpur
Bharatpur, Rajasthan, India

Surabhi Agrawal
Lecturer, Department of CSE&IT
Government Engineering College Bharatpur
Bharatpur, Rajasthan, India

***Abstract*:** In these days embedded systems can seen everywhere around us ranging from small electronic gadgets to big industrial machine. Embedded systems are microcontroller based systems used for performing specific task. In this work we are proposing methodology to develop embedded system with Atmel ATmega2560 Microcontroller (MCU) using Embedded C for programming. ATmega2560 is a low power CMOS 8-bit MCU of 100 pins which uses 86 pins out of 100 pins as I/O pins as user interface which makes it more feasible to complex embedded systems. This series features low ultra power consumption which making it best choice for IoT applications and it offers a widest range of devices which enables reuse of code and knowledge with different projects and software tools.

***Keywords*:** ATmega2560, CMOS, RISC, Atmel, Embedded C, AVR Studio, AVR Bootloader.

I. INTRODUCTION

ATmega2560 is low power CMOS 8-bit microcontroller having 100 pins. It is based on RISC architecture. ATmega2560 is suitable for the applications which require large amount of code to execute with performance up to 16 MIPS [7]. ATmega2560 microcontroller offers 256K byte in-system programmable flash memory with read-while-write capability which provides facility to reprogrammed in-system with ISP serial interface [2][5][7]. It also provides 4K byte of EEPROM, 8K byte SRAM and 32 general purpose registers. ATmega2560 microcontroller provides liberty to use 86 pins out of 100 pins as general purpose I/O pins to connect various peripherals. Availability of 86 pins as I/O pins make ATmega2560 microcontroller a feasible microcontroller to complex embedded systems. It has six flexible timer / counter with compare mode, a PWM generator, a 16 channel 10 bit ADC (analog to digital converter) and four USARTs interface. ATmega2560 is supported with Atmel QTouch library which enable it to provide capacitive touch input as well as sliders and wheels functionality [5][7]. Atmel provide an integrated development environment (IDE) (Atmel Studio / AVR Studio) and a strong library support to make easy and rapid development of system on ATmega2560. In our work we use AVR Studio 4.7 with embedded C to make program [4].

II. KEY FEATURES OF ATMEGA2560

1. Wide Support — ATmega series of microcontroller from Atmel offers a widest range of supporting devices in terms of memories, pin counts and peripherals, which enabling reuse of code and knowledge across different projects [1].
2. Ultra-low power technology — ATmega series of microcontroller from Atmel feature ultra-low power consumption and various selectable low-powers sleep modes that make it ideal choice for battery-powered applications [1].
3. High integration — ATmega series of microcontroller from Atmel feature on-chip Flash, SRAM, SPI, TWI (I^2C), and USART, internal EEPROM, USB, CAN, and LIN, watchdog timer, a choice of internal or external precision oscillator, and general-purpose I/O pins, which simplifying the design of project [1].
4. Rapid development — Atmel provides a wide range of software tools and a strong library support with powerful in-system programming and on-chip debug to simplifies production line programming and field upgrades [1].
5. IoT ready — ATmega series of microcontroller from Atmel are among the best microcontrollers in the world when it compared to power consumption, making them a best choice for IoT applications [1].

III. BLOCK DIAGRAM OF ATMEGA2560 MICROCONTROLLER

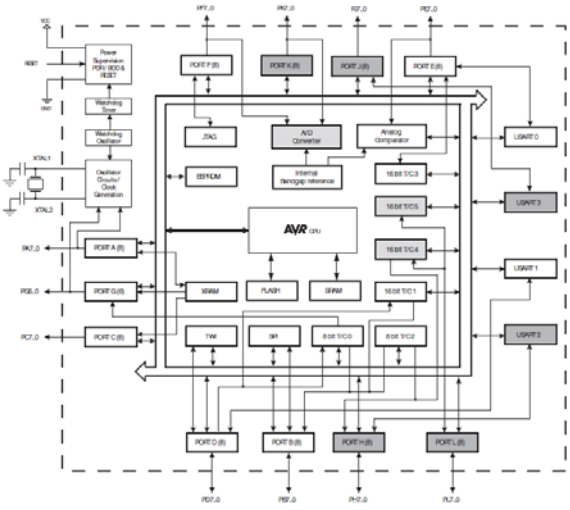

Figure 1:- Block diagram of ATmega2560 Microcontroller

IV. PROGRAMMING WITH ATMEGA2560 MICROCONTROLLER

ATmega2560 can be programmed with embedded C language. In our work we use AVR Studio 4.7. Before start programming we should concentrate on working of embedded system designed with ATmega2560. All the peripherals which we use in the system are connected with pins of ATmega2560 microcontroller therefore first we should understand how to use pins of ATmega2560 microcontroller. Out of 100 pins 86 pins are available for input output operation. 86 pins of ATmega2560 are grouped into 11 ports in which 10 ports are 8-bit port (named as A to F, H, J, K, L) and one 6-bit port (named as Port G). Any pin of ATmega2560's can be configured individually as either input pin or output pin depending the on value of DDRx register associated with that particular port [4][6].

V. REGISTERS FOR USING PINS (PORTS) OF ATMEGA2560 MICROCONTROLLER

All the ports of ATmega2560 microcontroller have three associated I/O registers called DDRx, PORTx, and PINx. Where x is port name A to L except I

A. **DDRx Register:** it called data direction register. It is used to set a port or pin as either input port or output port [4]. For example

DDRA = 0xF0; // this value in DDR register of port A sets upper nibble of port A as output pin and lower nibble as input pins;

For a particular pin, value 1 sets it as output pin and value 0 sets it as input pin.

B. **PORTx Register:** it is called port drive register. If the port is configured as output port PORTx sends the corresponding value to the pins configured as output pins of the associated port [4]. For example

DDRA = 0xFF; // this value in DDR register of port A sets all bits of port A as output pins;

PORTA = 0xF0; // this value in PORT register of port A sets upper nibble to value 1 and lower nibble to value 0;

C. **PINx Register:** it is called port pin register. For reading the input value from a port we read value of PINx register [4].

DDRA = 0x00; // this value in DDR register of port A sets all bits of port A as input pins;

x=PINA; // by this statement we read input value from port A using PINA register of port A into the variable x.

Note: all values given to registers or variables are in hexadecimal form (i.e. 0xFF)

VI. BASIC STRUCTURE OF EMBEDDED C PROGRAM FOR ATMEGA2560

Before to start writing a embedded C program for a ATmega2560 microcontroller we need to a basic layout of a program structure in mind. Here we discuss some basic steps to write a program in embedded C [4][6].

A. First; header file avr/io.h must be included, which enables input / output functionality of ATmega2560.
B. Ports of ATmega2560 must be configured as input or output port before to start using them. Only after configuration of ports they can be used to write to or reading from.
C. Every embedded C program must have main function which is starting point for execution of whole program.
D. An infinite loop should be there in the program so that device could wait for user commands until device is turned off by the user. All the functions which are repeatedly needed for every command of user should be called from inside an infinite loop.

VII. EXAMPLE PROGRAM

/** Description: This program is aimed to turn on upper four LEDs connected on port J when sense pressing of a push button switch.

* Functions: main(),port_init(),led_on(),led_off().

*/

// Note : You may write more functions depending on requirement.

#include <avr/io.h> //for including functionality of input / output at pins of ATmega2560 MCU

#include <util/delay.h> //for using time functions

/*Function name: port_init

Input: none

Output: none

Logic: Code to initialize desired I/O ports*/

void port_init(void)

{DDRJ = 0xF0; //PORTJ upper nibble set as output DDRE = DDRE & 0x7F; //PORTE 7 pin set as input of port E; where a switch button is conneted

PORTE = PORTE | 0x80; //PORTE7 internal pull-up enabled.}

/*Function name: led_on

Input: none

Output: none

```
Logic: Code to turn on bar led*/

void led_on(void)

{PORTJ = 0xF0;

// TURN ON upper four LEDs connected on port J.}

/*Function name: led_off

Input: none

Output: none

Logic: Code to turn off bar led*/

void led_off(void)

{PORTJ = 0x00;  // TURN OFF all LEDs connected on port J}

/*Function name: main

Input: none

Output: none

Logic: Program Execution starts from this function.*/

int main()

{port_init();// initialize portswhile(1)     // a continuous loop. It waits to press the switch button until device is turned off.

{if((PINE & 0x80) == 0x80) //switch is not pressed

{led_off(); //Turn off leds  }

else

{led_on(); //Turn on leds} }}
```

VIII. COMPILATION AND TESTING OF PROGRAM

In our work we use AVR Studio 4.7 IDE for programming purpose and AVR Bootloader (from next robotics) to burn program to microcontroller using a USB cable. After writing program in AVR Studio, code should be compiled by pressing Alt+F7 key or by selecting compile submenu in build menu of AVR Studio. In compilation process only syntactical error are can be found. Logical errors can be tested only after burning program code to microcontroller. After compilation if there is no error the code should be converted into hex format (because code must be in hex format; only hex format is understandable by microcontroller). For converting code in hex format press F7 key or select Build submenu in build menu of AVR Studio. After pressing F7 key a hex file is generated [4].

Before burning hex code in microcontroller; microcontroller must be in boot or burn mode, connect USB cable to microcontroller and open AVR Bootloader. Select hex file which is generated form program code and write it to the microcontroller. After burning hex file to microcontroller reset microcontroller and test the functionality of program [4].

IX. APPLICATIONS OF ATMEGA2560 MICROCONTROLLER

ATmega2560 microcontroller has wide range of applications. Here some examples of applications are listed below [3]:

- Automotive.
- Building Automation.
- Home Appliances.
- Home Entertainment.
- Industrial Automation.
- Internet of Things.
- Smart Energy.
- Mobile Electronics.

X. CONCLUSION

Since the embedded systems are very popular in these days these can be range from small home appliance to a big aircraft. Embedded C is efficient (in the form of big library support and low level working) and easy (in the form of syntax) language for developing embedded systems. In carried out work, we use ATmega2560 microcontroller which suitable for a wide range of applications as it has a high integration level on single chip which reduce need of external devices and strong library support from Atmel, for high level functions and peripheral support make programming easy with ATmega2560.

XI. REFERENCES

[1]. Atmel, megaAVR Microcontrollers [online] Available: http://www.atmel.com/products/microcontrollers/avr/megaavr.aspx [Accessed June 05, 2017].

[2]. Nex Robotics, ATMEGA2560 Microcontroller Socket [Online] Available: http://www.nex-robotics.com/products/microcontroller-development-boards/atmega2560-microcontroller-socket.html [Accessed June 6, 2017].

[3]. Atmel, Applications [Online] Available: http://www.atmel.com/applications/automotive/default.aspx?src=parent [Accessed June 6, 2017].

[4]. Nex Robotics, Fire Bird V ATMEGA2560 Software Manual V1.00 15-08-2012.

[5]. Nex Robotics, Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.

[6]. AVR Tutorials [Online] Available: http://www.avr-tutorials.com/digital/digital-input-output-c-programming-atmel-avr-8-bits-microcontrollers [Accessed June 8, 2017].

[7]. Microchip, ATmega2560 [Online] Available: http://www.microchip.com/wwwproducts/en/ATmega2560 [Accessed 8 June 2017].