computers

MDPI

Type of the Paper (Article)

Designing Microcontroller module with target to use AVR/PIC assembler

Veselin Stanchev [1]

[1] Affiliation 1; vrstanchev@tu-plovdiv.bg
* Correspondence: Veselin Stanchev (e-mail: vrstanchev@tu-plovdiv.bg).

Abstract: Assembly languages are very useful for programming of micro-controllers, microprocessors, embedded devices, real-time operating systems. The most natural way for micro-controller programming is to use Assembly languages. For different types of micro-controller families - AVR/PIC-based, there are different assembly languages. For AVR-based micro-controllers such as - ATtiny13,ATtiny85,ATMega32u4 different assembly languages can be used – AVR Assembler or GNU Assembler. Micro-controllers can be used for micro-controller modules design. This paper describes design and programming of micro-controller module which uses both AVR-based and PIC-based micro-controller. This module can be used for educational purposes. It shows communication between its micro-controllers.

Academic Editor: Veselin Stanchev

Received:
Revised:
Accepted:
Published:

Keywords: PCB; MCU; AVR/PIC; Assembly language; Micro-controller module

1. Introduction

Usage of free assembly languages is very important for understanding Computer Architectures. Free assembly languages can be used for university purposes. To explain usage of them for micro-controllers, micro-controller boards can be designed using open-source EDA software such as: Fritzing or LibrePCB. On single micro-controller board can be used micro-controllers together. They are several variants for that – usage of PIC-based micro-controllers, usage of AVR-based micro-controllers or AVR/PIC micro-controllers together.Target of the current project is to use AVR assembler anPIC assembler through custom communication protocol. There are requirments for hardware and software part of the project. ATtiny85 and PIC10F320 are chosen.

For software part of the project are used tools such as: Git, GNU Make, GNU GCC, GNU PIC Utils, avra. Git is used for project version-control. GNU Make is used for automatization of software build. Software of the project is also produced as static Assembly library. Hardware part is created by using of LibrePCB. LibrePCB is used for schematic design, footprints of micro-controllers and layout of the micro-controller module.

[1]

2. Materials and Methods

Project orientation is fully practical. The micro-controller module can be used for university purposes. For realization of the project are used open-source tools. Because of that project can be upgraded easily. The project can be used for: assembly language labs, micro-controller labs, understanding of micro-controller communication, understanding of AVR/PIC programming.

1) **Existing micro-controller modules ANAVI MACRO PAD 2, ANAVI LIGHT CONTROLLER, Raspberry Pi Pico and AVR64DD32 Curiosity Nano. For each of them are analyzed: supported micro-controllers, supported communication protocols, supported interfaces, supported assembly languages.**

   Example: ANAVI MACRO PAD 2 and ANAVI LIGHT CONTROLLER [2],[3].

FIGURE 1. ANAVI MACRO PAD 2

FIGURE 2. ANAVI LIGHT CONTROLLER

Requirements of the project are hardware-based and software-based.

2) **For hardware-based requirements: AVR/PIC-based micro-controllers has to be chosen. ATtiny85 and PIC10F320 are chosen.**

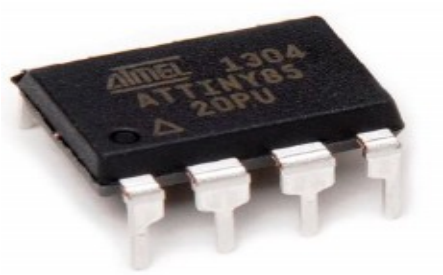


FIGURE 3. ATTiny85

FIGURE 4. PIC10F320

Micro-controller's features are analyzed. ATTiny85 has features such as: 8kb flash memory, 8-bit timer, communication thought universal serial interface, communication thought I2C. PIC10F320 has features such as: 128 bytes of memory, 8-bit timer, communication thought host and micro-controller. [4],[5]

For design of micro-controller module LibrePCB is chosen. AVR/PIC-based micro-controller's symbols are created. Footprints for them are created.

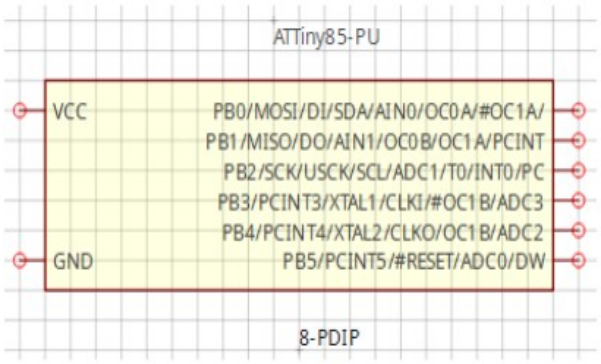


FIGURE 5. Symbol of ATTiny85

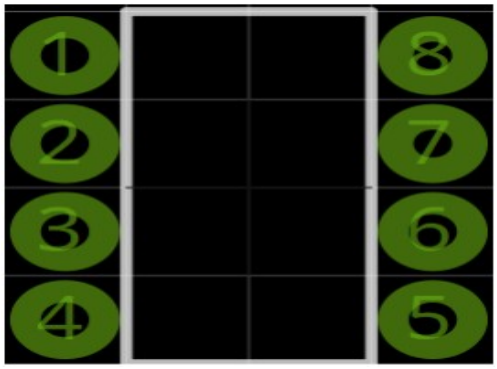


FIGURE 6. Footprint of ATTiny85

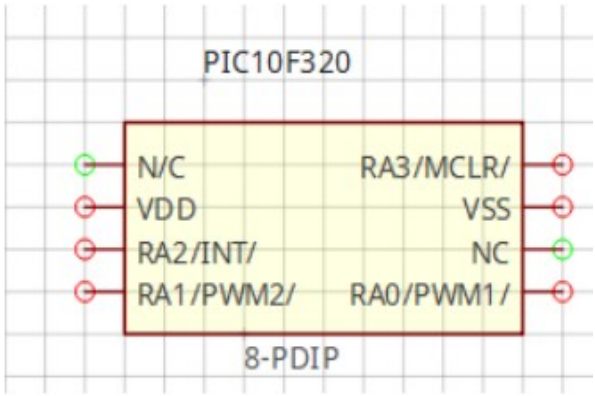


FIGURE 7. Symbol of PIC10F32

FIGURE 8. Footprint of PIC10F320

USB interface is chosen for power and programming. So micro-controller module block diagram is created.

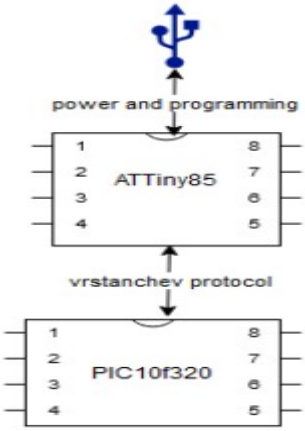


FIGURE 9. Micro-controller module block diagram

[1]

Due to chosen micro-controllers , interfaces and block diagram micro-controller module's schematic is created. Components are connected directly thought wires or net labels

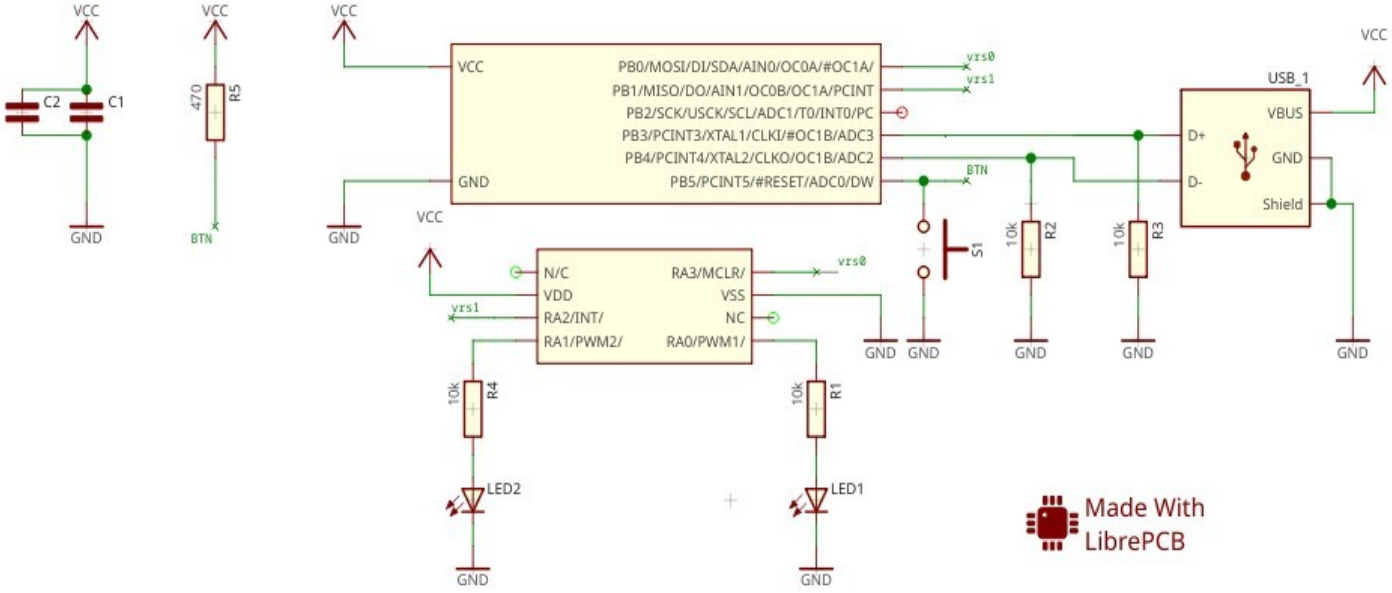


FIGURE 10. Micro-controller module schematic [1]

Interface	Connectivity	Used Pins
USB	Attiny85 ⇔ USB	PB3,PB4
Communication protocol vrstanchev	Attiny85 ⇔ PIC10F320	PB0, PB1 → RA2, RA3

FIGURE 11. Interfaces and used pins table [1]

PIN Type	PIN
output	PB0
output	PB1
input	RA2
input	RA3

FIGURE 12. Pin-type table [1]

Component	Used Pins
ATTiny85	PB0, PB1
PIC10F320	RA2, RA3
LED1, LED2	RA0, RA1

FIGURE 13. Components and used pins [1]

3) For software-based requirements: Assembly languages for ATtiny85 and PIC10F320 has to be chosen. AVR assembly and PIC assembly languages are chosen. There are given examples for micro-controllers intructions. Custom vrstanchev protocol is implemented. Code for ATTiny85 and PIC10F320 is implemented thought functions. It is chosen for micro-controllers communication. Logic of communication is shown. Build of the code is automated thought GNU Make. Function for ATTiny85 is compiled via avra compiler. Function for PIC10F320 is compiled via gpasm compiler. [1]

FIGURE 14. Schematic of vrstanchev protocol functions [1]

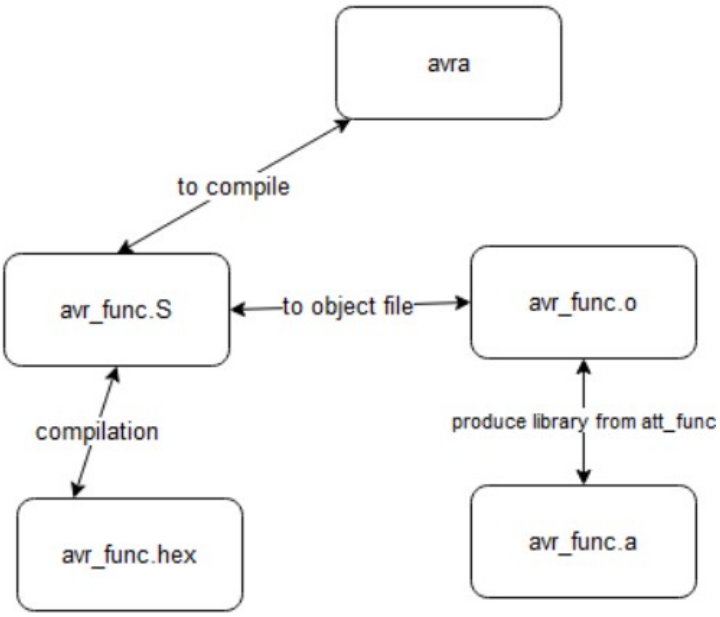


FIGURE 15. Schematic of att_func() Compilation [1]

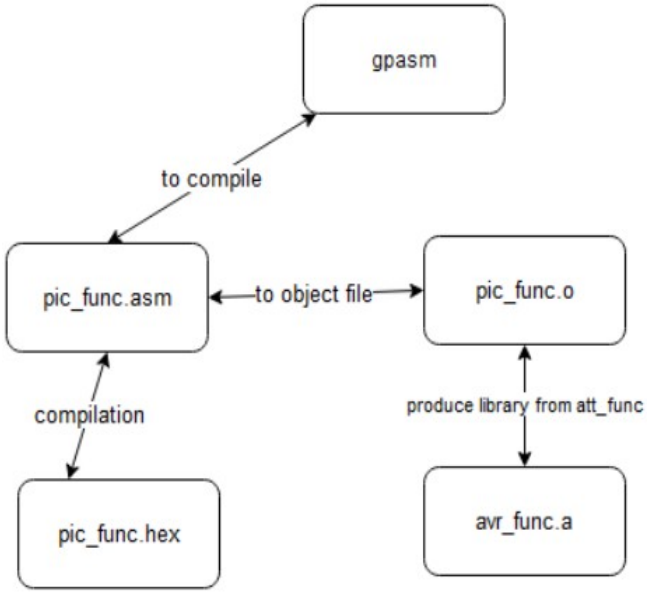


FIGURE 16. Schematic of pic_func() Compilation [1]

Build of att_func()	**Build of pic_func()**
compile:att_func.S avra att_func.S att_func.o:att_func.S avra -o att_func.o att_func.S	compile:pic_func.asm gpasm pic_func.asm pic_func.o:pic_func.asm gpasm -c pic_func.asm

FIGURE 17. GNU Make Build scripts [1]

Part of source code of the ATTiny85 assembly function:	***Part of source code of the PIC10F320 assembly function:***	***Part of source code of the vrstanchev protocol:***
att_func: *.include "tn85def.inc"* *.equ pb0_out=0b0000001* *.equ pb1_out=0b0000010* *.cseg* *.org 0x00* *sbi r16, pb0_out* *sbi r17, pb1_out*	.include "pic10f320.inc" .org 0 pic_func: CLRF PORTA MOVLW B'00000100' MOVLW B'00001000' MOVLW B'00000100' MOVLW B'00001000'	vrstanchev: mov B'00000100', r0 mov B'00001000'r1 mov B'00000100',r3

FIGURE 18. Parts of the source code [1]

4) Assembly language functions are compiled also as static library. Static libraries are archives of compiled object files(with extension .o). For ATTiny85 function achiever is GNU ar and for PIC10F320 function achiever is GNU PIC Library (gplib). Static library schematic is given below:

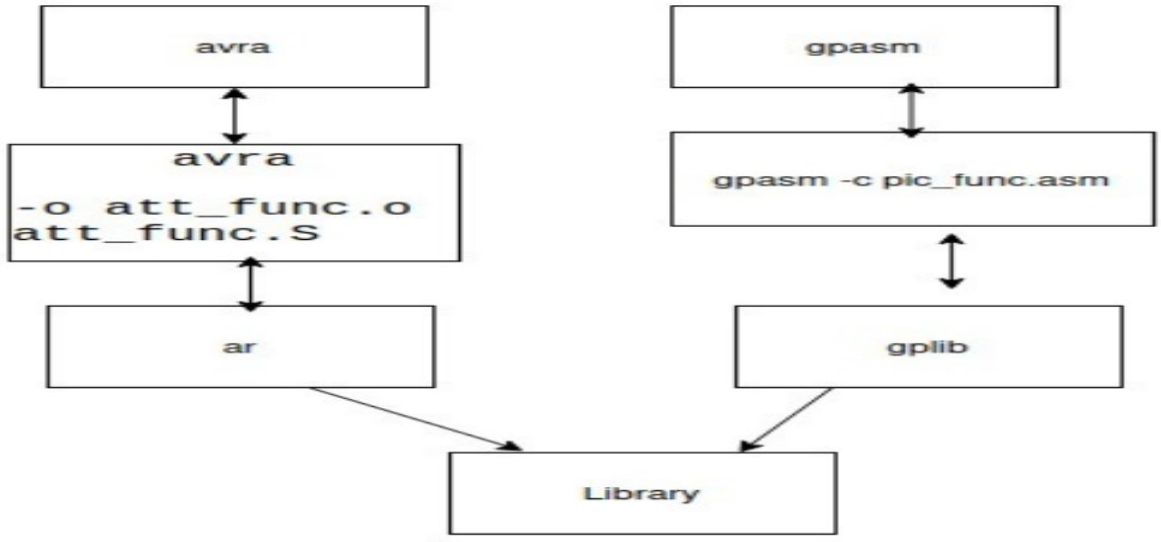
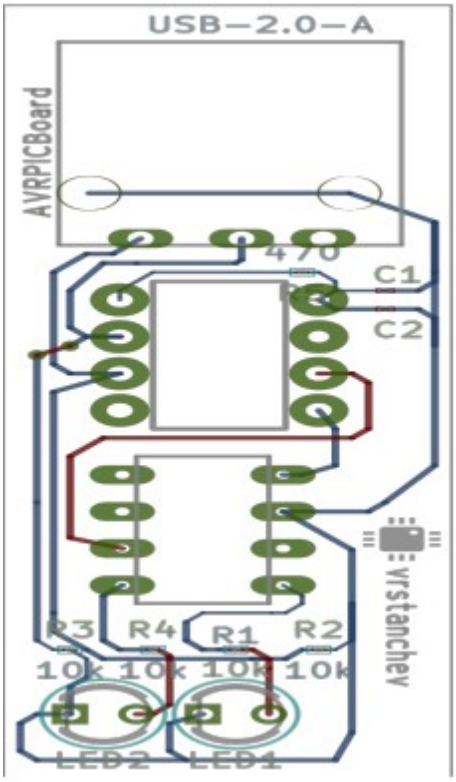


FIGURE 19.Static library schematic

3. Results

Micro-controller module powered by ATTiny85 and PIC10F320 is designed. LibrePCB is used to design the module. Because of using tools that are open-source or are part of the GNU project, micro-controller module's features are easy to been upgraded. For programming of ATTiny85 is used AVR Assembly and for PIC10F320 is used PIC Assembly. Micro-controllers communicates each other thought implemented vrstanchev protocol. The protocol is available also as static Assembly library.



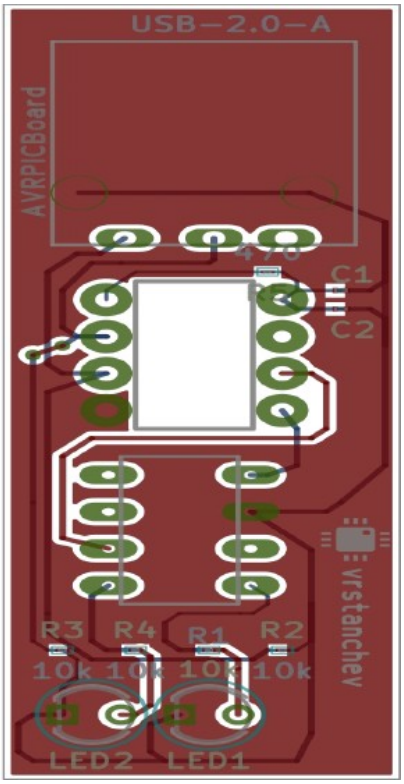


FIGURE 20. Micro-controller module board layout [1] **FIGURE 21. Background noise of the module is reduced**

4. Conclusions

Hardware and software requirements of the project are completed.The target of the project is complete. Features of the micro-controller module to been added: interfaces for main module, additional modules for main module, different releases of main module, micro-controller module powered by AVR-AVR or PIC-PIC micro-controllers.

References

Reference to a Master Degree Thesis:

[1] V.Stanchev "Designing Microcontroller module with target to use AVR/PIC assembler" Master Degree Thesis.

Reference to a technical datasheet:

[2] L.Anavi *ANAVI Macro Pad 2 DataSheet*., Plovdiv, Bulgaria.

Reference to a technical datasheet:

[3] L.Anavi ANAVI Light Controller DataSheet, Plovdiv, Bulgaria.

Reference to a technical datasheet:

[4] Atmel, ATTiny85 Datasheet.

Reference to a technical datasheet:

[5] Microchip, PIC10F320 Datasheet.