

MIKROKONTROLER ATMEGA BERBASIS CODEVISION AVR (PENDAHULUAN)

Mikro-(kontroler vs prosesor)

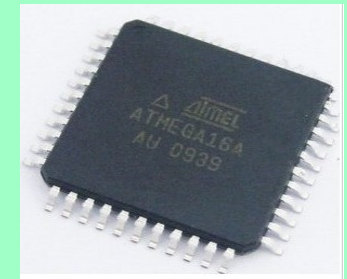
Fitur	Mikroprosesor	Mikrokontroler
Memori RAM - Keberadaan	Eksternal	Internal
Memori RAM - Kapasitas	Besar (Giga Bytes)	Kecil (Kilo Bytes)
Memori ROM - Keberadaan	Eksternal	Internal
Memori ROM - Kapasitas	Besar (Tera Bytes)	Kecil (Mega Bytes)
CPU - Kecepatan	Tinggi (Giga Hertz)	Rendah (Mega Hertz)
Tujuan Penggunaan	Memproses data yang kompleks:	Mengendalikan suatu proses:
	> Aplikasi <i>games</i>	> Mengendalikan ROBOT
	> Web server	> Alat ukur jarak
Operating System (OS)	Wajib	Tidak wajib
Contoh IC	AMD Athlon	ATMEL AT89S51
	Intel Pentium	ATMEL ATmega8535
		ATMEGL ATmega16A
		ATMEL ATtiny2313
		PIC

Mikrokontroler AVR

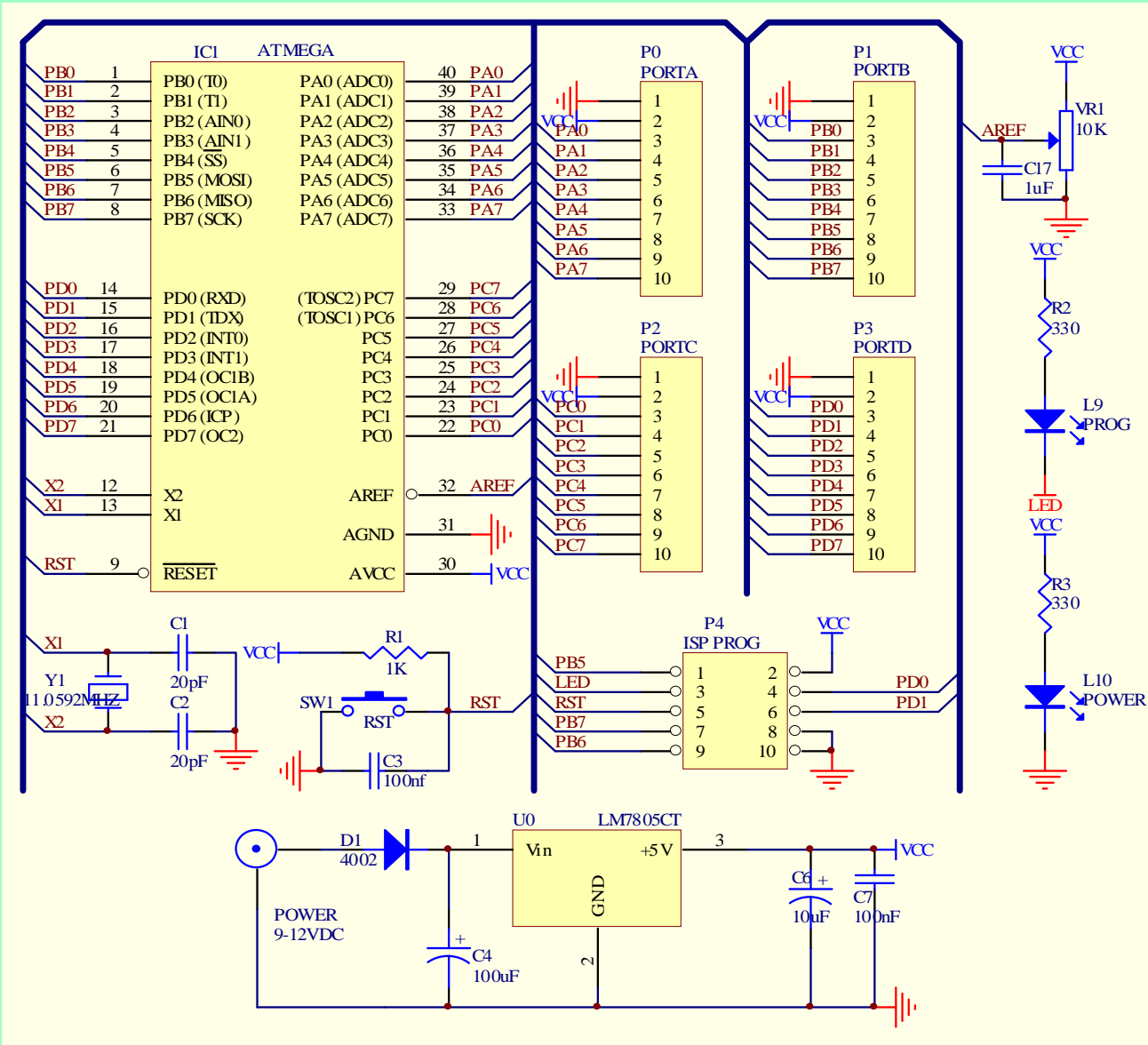
- Perusahaan: ATMEL
- AVR = ?
- Divergensi:
 - ▣ ATmega
 - ▣ ATtiny
 - ▣ Atxmega
 - ▣ Special Function

Mikrokontroler ATmega16A

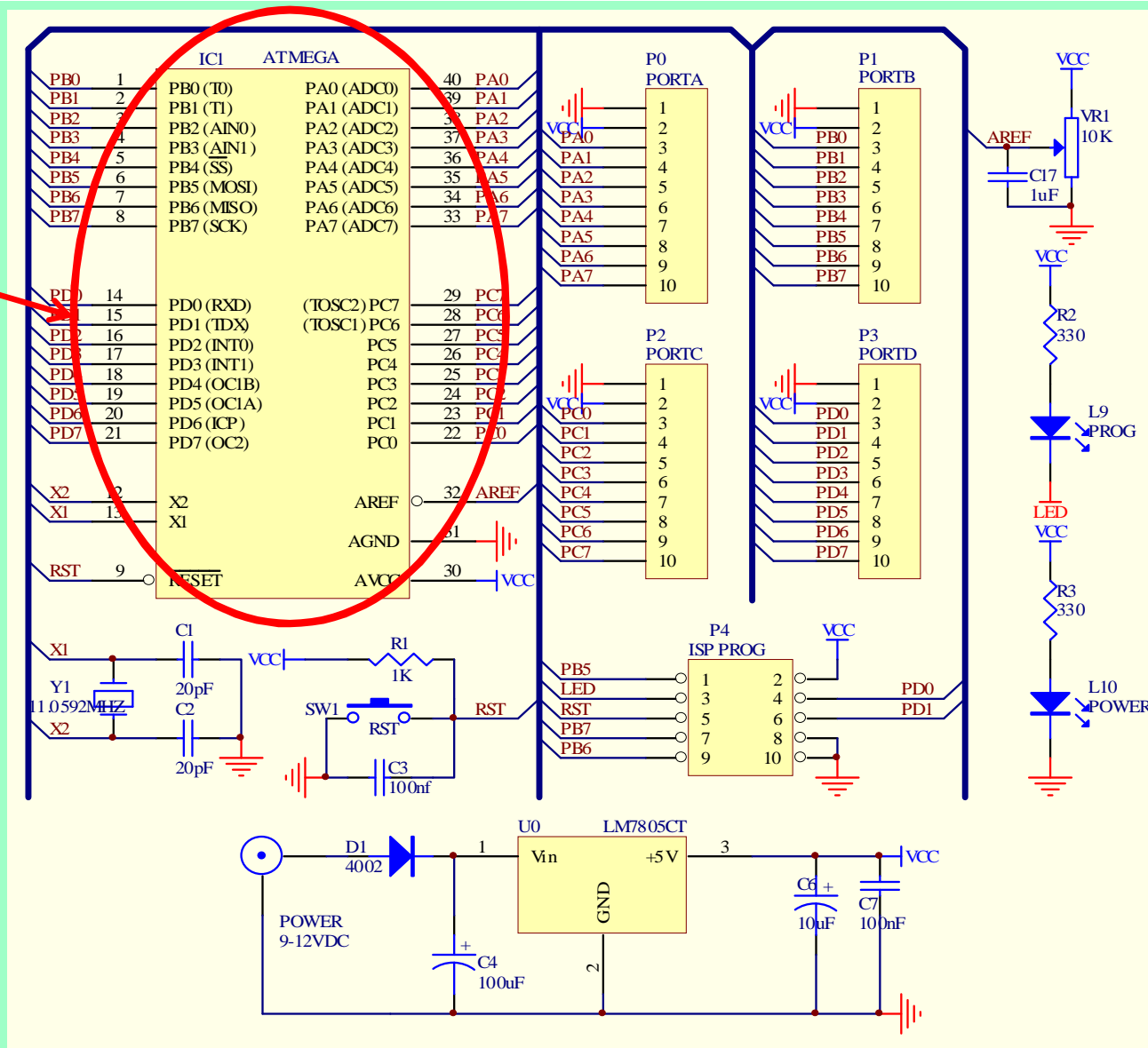
- High-performance, Low-power Atmel AVR **8-bit Microcontroller**
- Advanced RISC Architecture. Up to 16MIPS Throughput at 16MHz
- High Endurance Non-volatile Memory segments
 - ▣ 16KBytes of In-System Self-programmable Flash program memory
 - ▣ 512Bytes EEPROM
 - ▣ 1KByte Internal SRAM
- Peripheral Features
 - ▣ Timers: Two 8-bit Timer/Counters and One 16-bit Timer/Counter
 - ▣ 8-channel, 10-bit ADC
 - ▣ Serial Communication: Programmable Serial USART, Master/Slave SPI Serial Interface, Byte-Oriented Two Wire Interface.
- I/O and Packages: 32 I/O Lines. 40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF
- Operating Voltages: 2.7 - 5.5V
- Speed Grades: 0 - 16MHz



Sistem Minimum μ C. ATmega16A



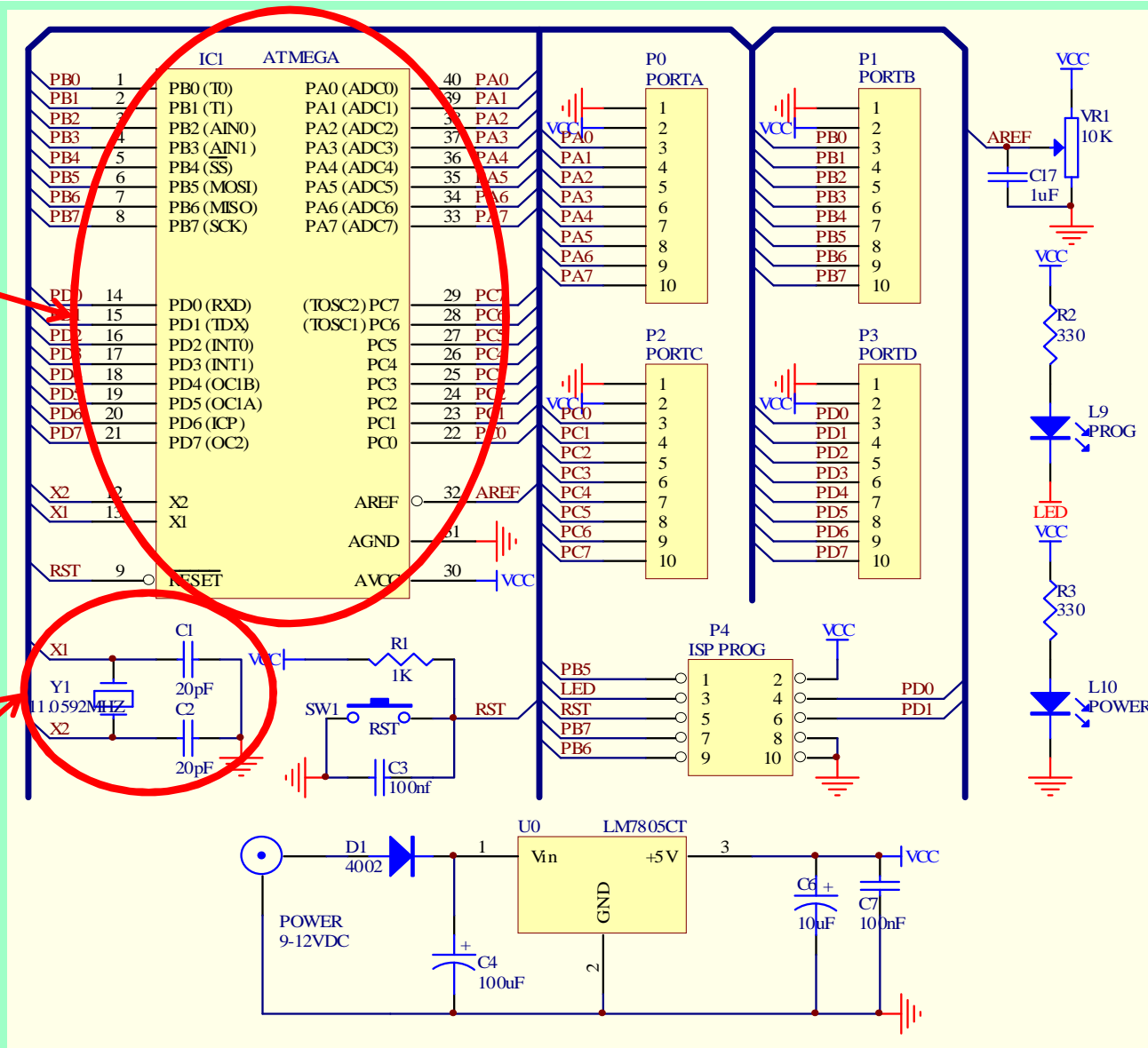
Sistem Minimum μ C. ATmega16A



Sistem Minimum μ C. ATmega16A

IC ATmega16A

Clock Source
(Crystal)

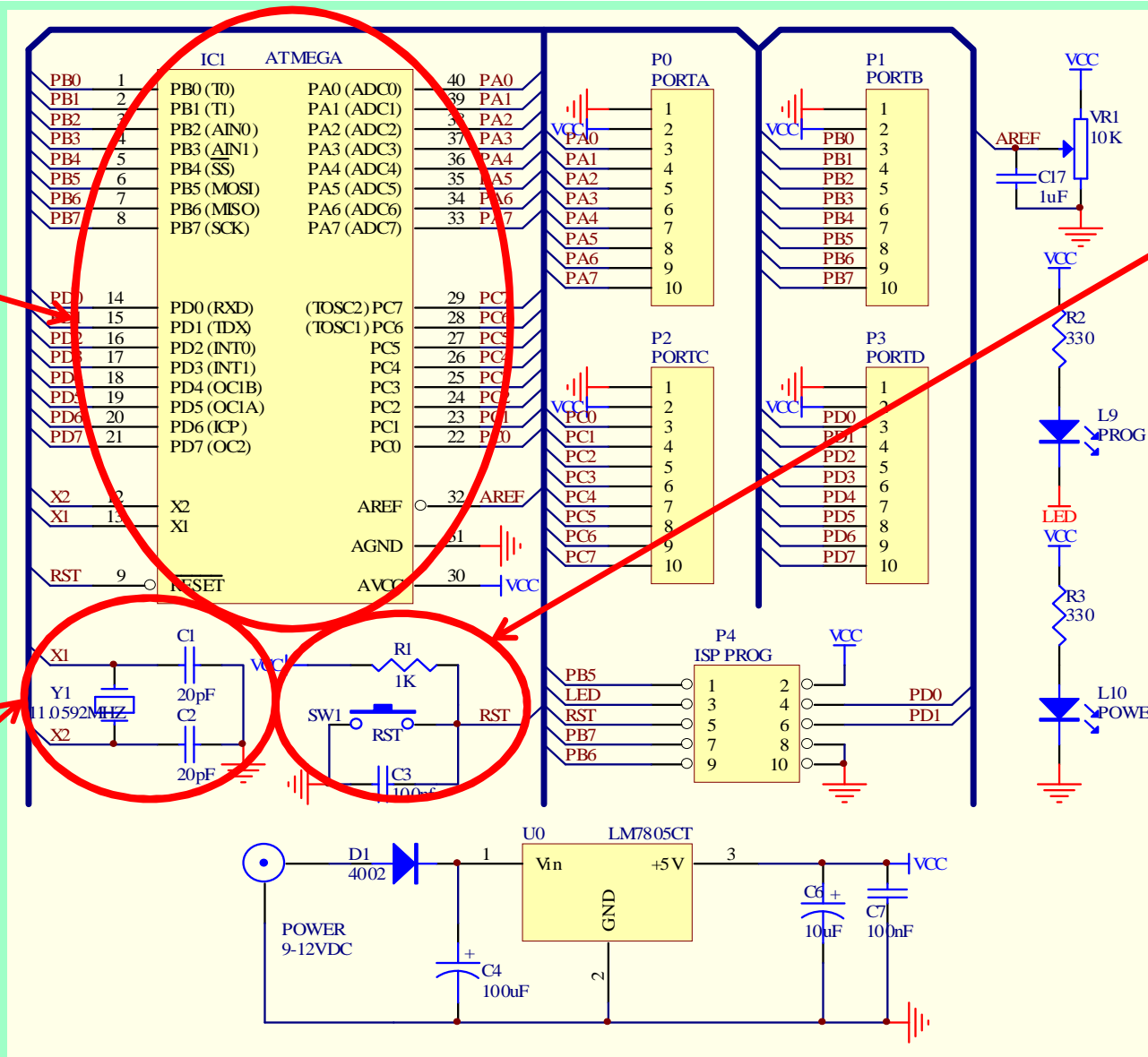


Sistem Minimum μ C. ATmega16A

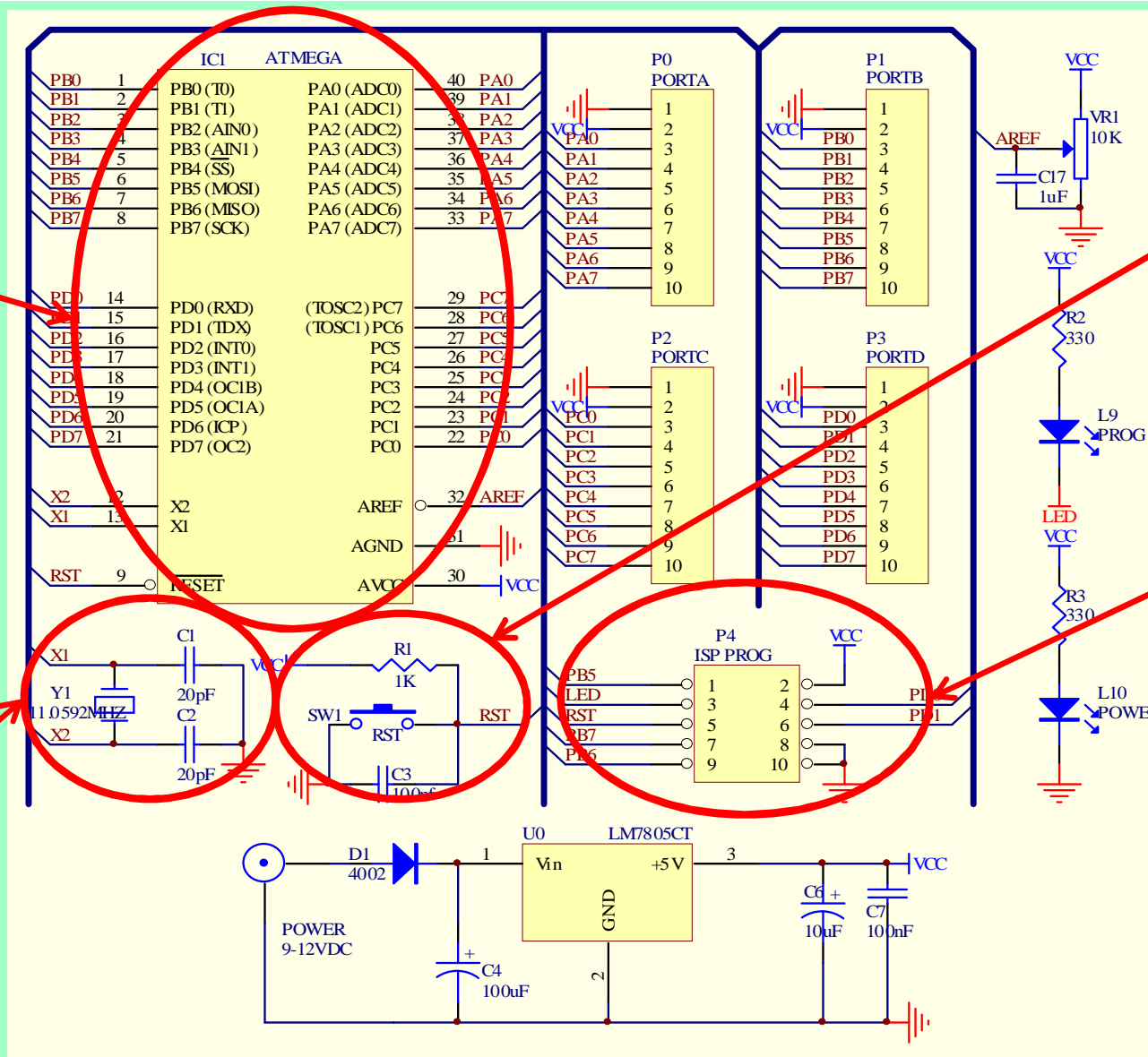
IC ATmega16A

Clock Source
(Crystal)

Reset System



Sistem Minimum μ C. ATmega16A



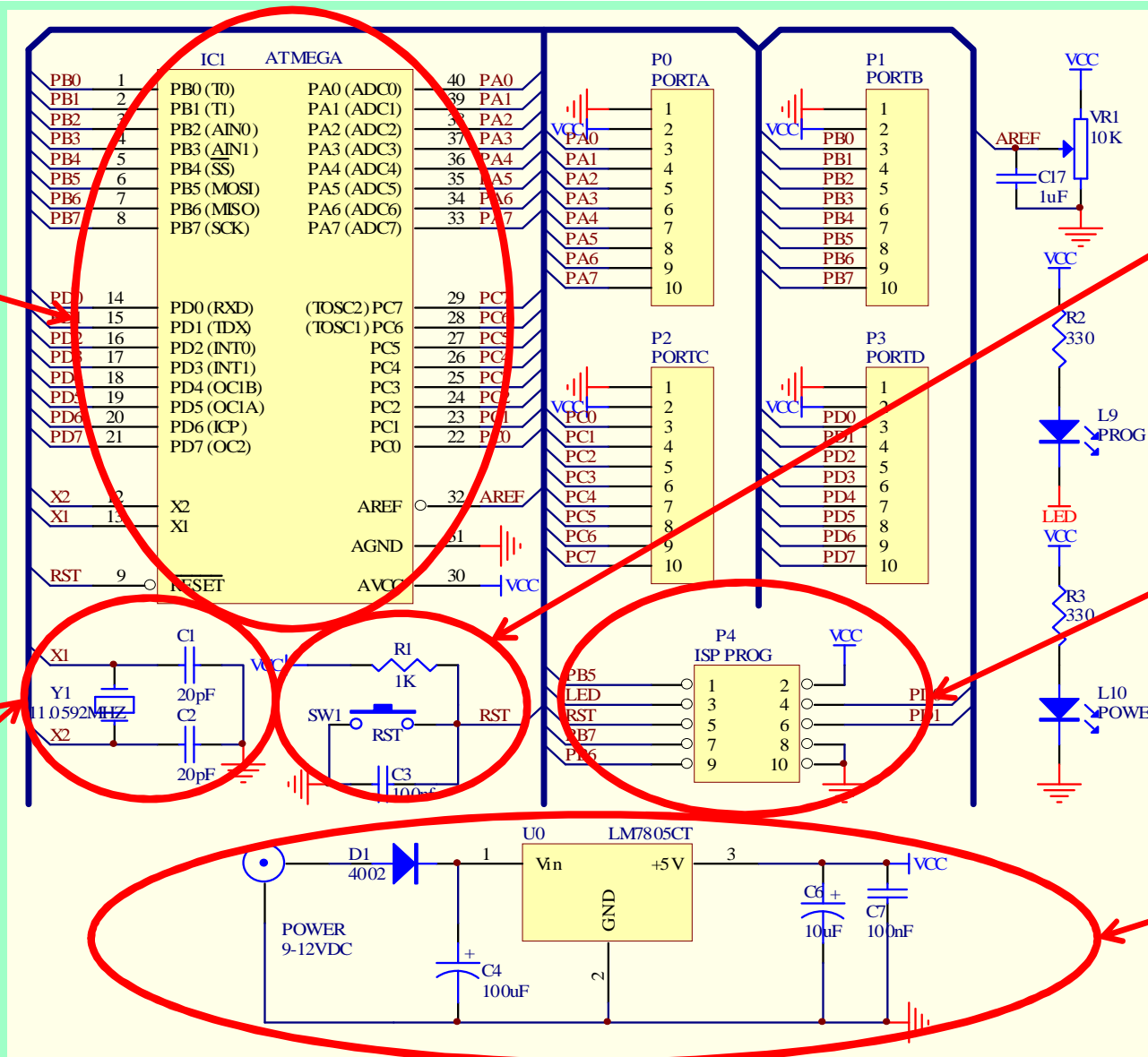
IC ATmega16A

Clock Source
(Crystal)

Reset System

In System
Programming

Sistem Minimum μ C. ATmega16A



IC ATmega16A

Clock Source
(Crystal)

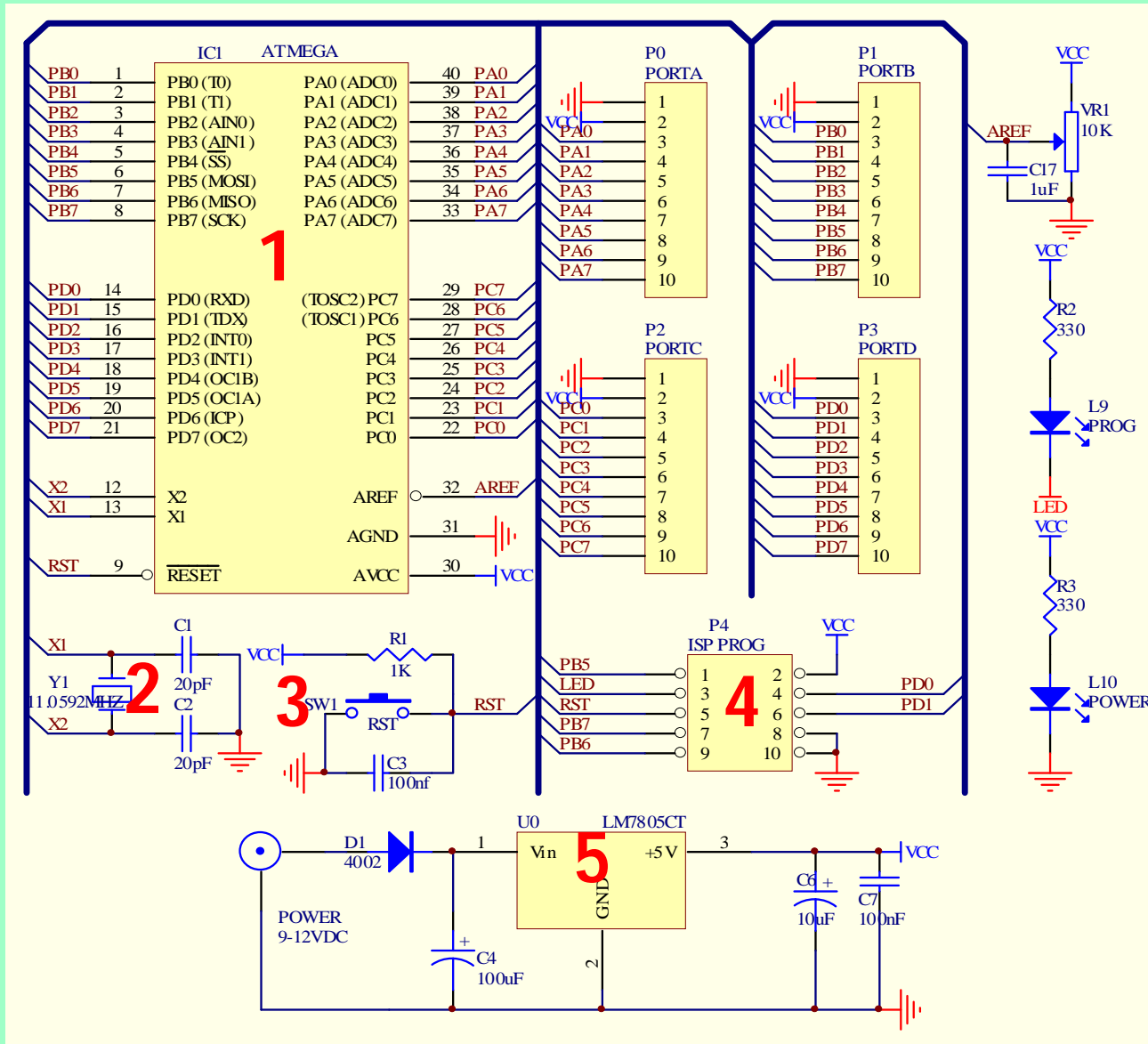
Reset System

In System
Programming

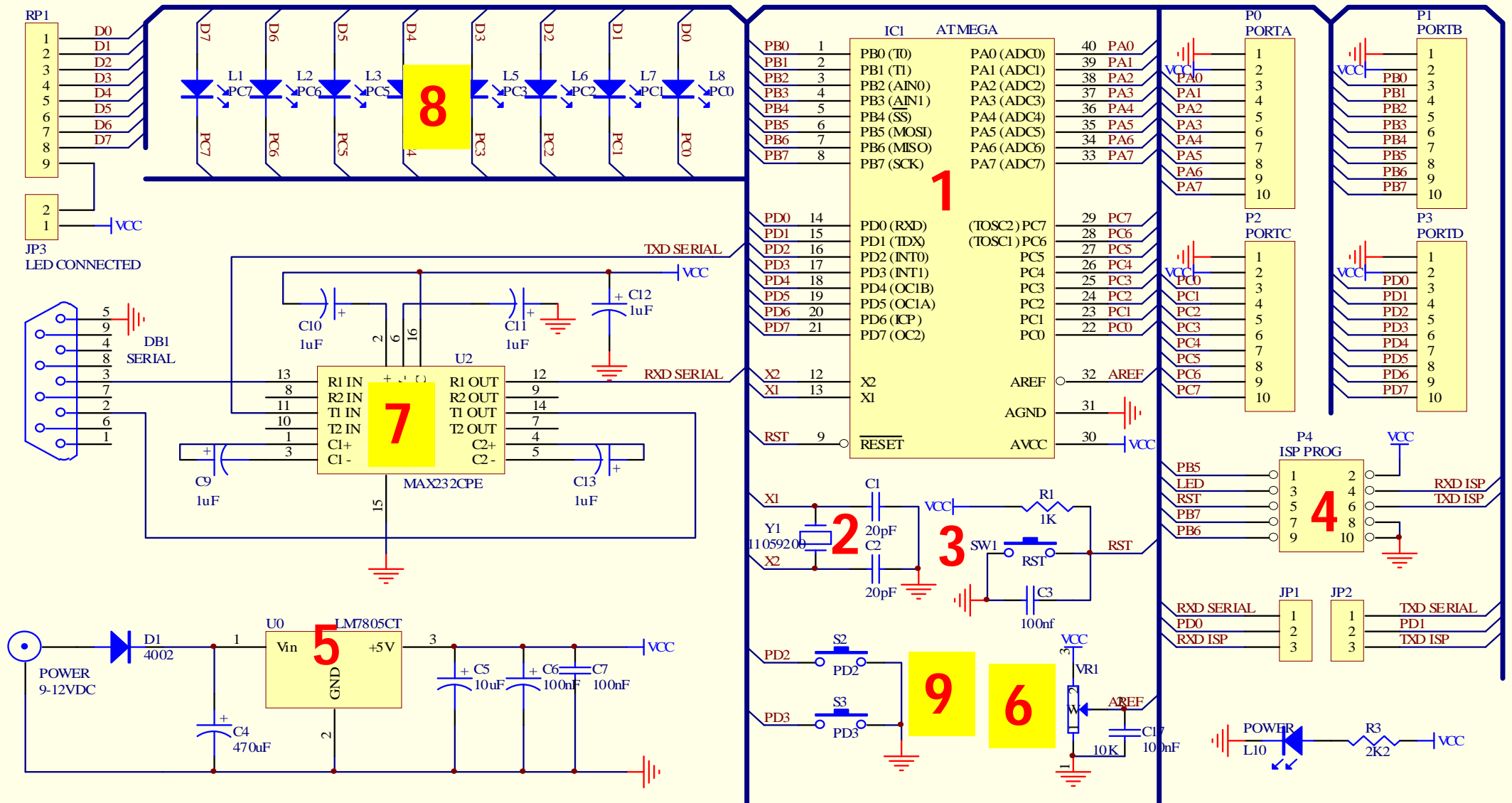
Power Supply

Sistem Minimum μ C. ATmega16A

1. ATmega16A
2. Crystal
3. Reset
4. ISP
5. Power

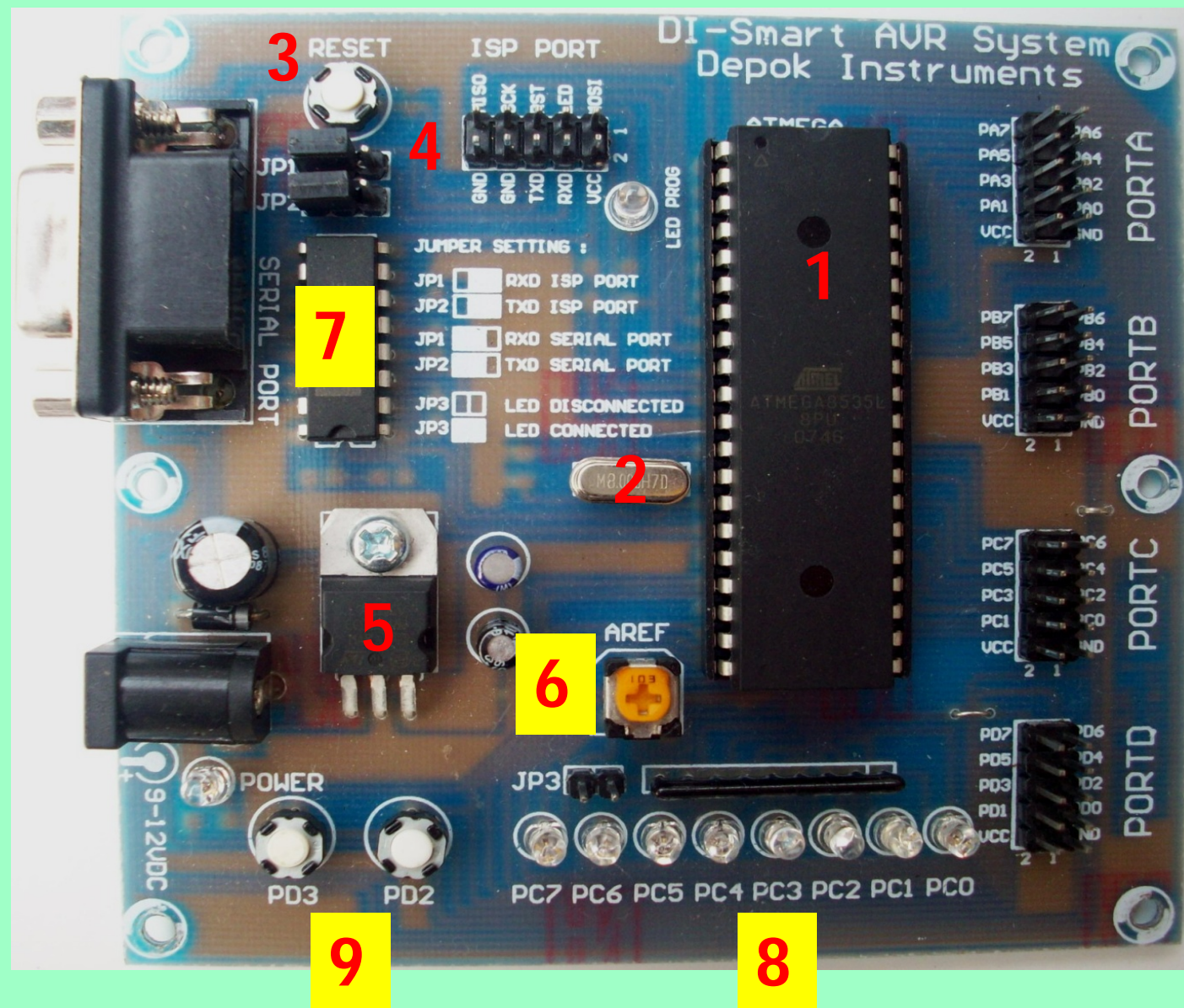


DI-Smart AVR.16 System (Sistem Plus μ C. ATmega16A)



DI-Smart AVR.16 System

1. ATmega16A
2. Crystal
3. Reset
4. ISP
5. Power
6. Vref ADC
7. RS232
8. LED Array
9. Push Button



Cara Memprogram μ C. ATmega16A

- ARDUINO
- BASCOM
- **Code Vision AVR**

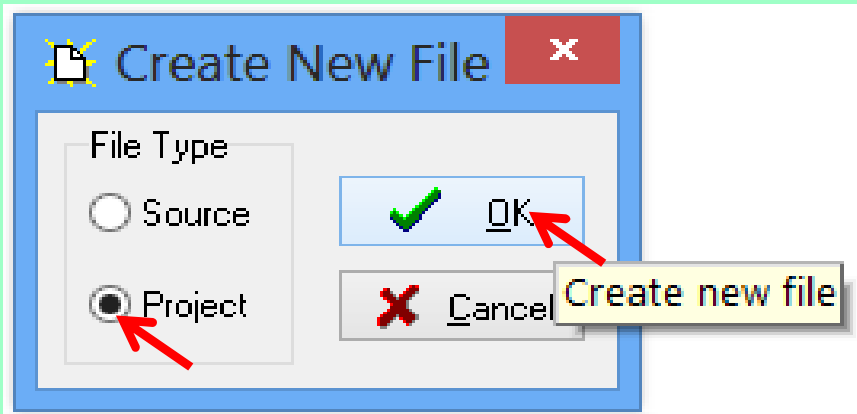
Code Vision AVR

- Perusahaan: HP InfoTech

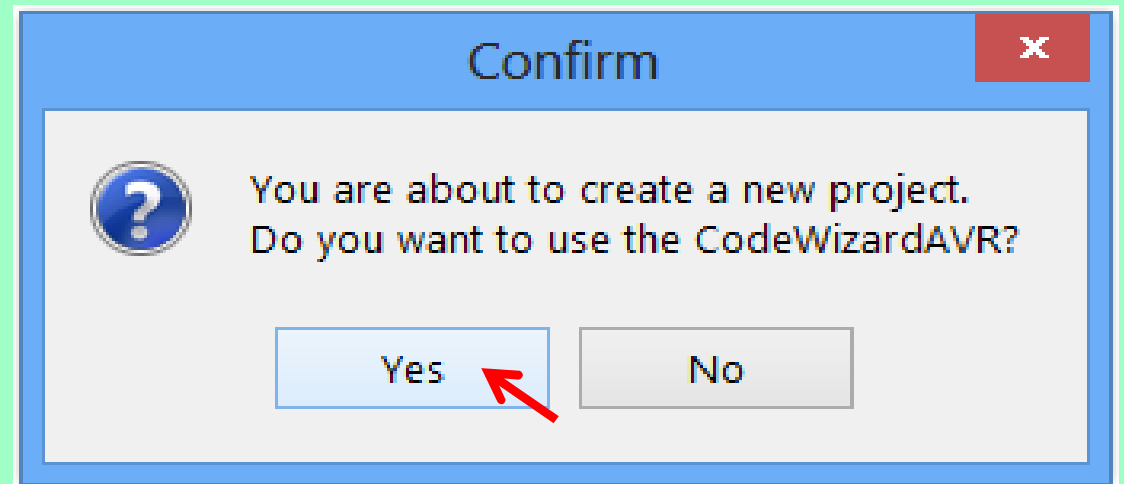


"File >> New" Code Vision AVR

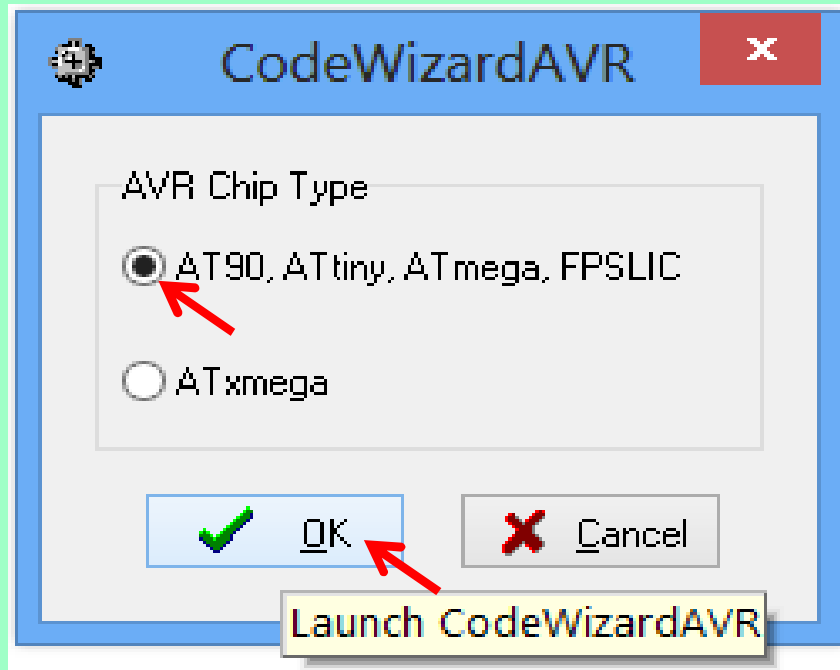
1



2



3



"Wizard" Code Vision AVR

USART	Analog Comparator	ADC	SPI
I2C	1 Wire	TWI (I2C)	
Alphanumeric LCD			
Bit-Banged		Project Information	
Chip	Ports	External IRQ	Timers

Chip:

Clock: MHz

—

Program Type:

- **USART = Universal Serial Async RX TX**
- **Analog Comparator**
- **ADC = Analog to Digital Converter**
- **SPI = Serial Peripheral Interface**

- **I2C = Inter-Integrated Circuit**
- **1 Wire = One Wire Interface**
- **TWI (I2C) = Two Wires Interface**

- **Alphanumeric LCD**

- **Bit-Banged**
- **Project Information**

- **Chip**
- **Ports**
- **External IRQ = Analog to Digital Converter**
- **Timers**

Pemrograman Tampilan LED (Setup "Wizard")

1

USART Analog Comparator ADC SPI
I2C 1 Wire TWI (I2C)
Alphanumeric LCD
Bit-Banged Project Information
Chip Ports External IRQ Timers

Chip: ATmega16
Clock: 11.059200 MHz
 Check Reset Source
Program Type: Application

2

USART Analog Comparator ADC SPI
I2C 1 Wire TWI (I2C)
Alphanumeric LCD
Bit-Banged Project Information
Chip Ports External IRQ Timers

Port A	Port B	Port C	Port D
Data Direction			Pullup/Output Value
Bit 0	Out	0	Bit 0
Bit 1	Out	0	Bit 1
Bit 2	Out	0	Bit 2
Bit 3	Out	0	Bit 3
Bit 4	Out	0	Bit 4
Bit 5	Out	0	Bit 5
Bit 6	Out	0	Bit 6
Bit 7	Out	0	Bit 7

3

File Program Edit Help

Generate program, save and exit

Pemrograman Tampilan LED (Hasil Setup "Wizard")

```
D:\PROGRAM CVAVR\P01_LED.c
Notes P01_LED.c *
31
32 // Input/Output Ports initialization
33 // Port A initialization
34 // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
35 // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
36 PORTA=0x00;
37 DDRA=0x00;
38
39 // Port B initialization
40 // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
41 // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
42 PORTB=0x00;
43 DDRB=0x00;
44
45 // Port C initialization
46 // Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
47 // State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
48 PORTC=0x00;
49 DDRC=0xFF;
50
51 // Port D initialization
52 // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
53 // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
54 PORTD=0x00;
55 DDRD=0x00;
```

- DDRX:
- 0 = Input
- 1 = Output

Pemrograman Tampilan LED (Kelap-Kelip)

```
□ #include <mega16.h>
□ #include <delay.h>
□ void main(void)
□ {
  □ PORTC=0x00;
  □ DDRC=0xFF;
  □ while(1)
    ■ {
      ■ PORTC = 0x00;
      ■ delay_ms(250);
      ■ PORTC = 0xFF;
      ■ delay_ms(250);
    ■ }
□ } //freestyle (File >> Save As >> Project >> Configure)
```

Pemrograman Tampilan LED (+Input *Push-Button*)

```
□ #include <mega16.h>
□ #include <delay.h>
□ void main(void)
□ {
  □ PORTC=0x00; DDRC=0xFF;
  □ PORTD=0x0C; DDRD=0x00;
  □ while(1)
    ■ {
    ■ //if, else if, else
    ■ if(PIND.2==0) PORTC = 0xF0; //Input bit = PINX.Y
    ■ else if(PIND.3==0) PORTC = 0x0F;
    ■ else PORTC = 0xFF;
    ■ }
□ }
```

E-BOOK DINS

- <http://depokinstruments.com/category/3-e-book/>