Zbuduj sobie gadżet
czyli mikrokontrolery AVR pod Linuxem

Zimowisko Linuxowe 2011

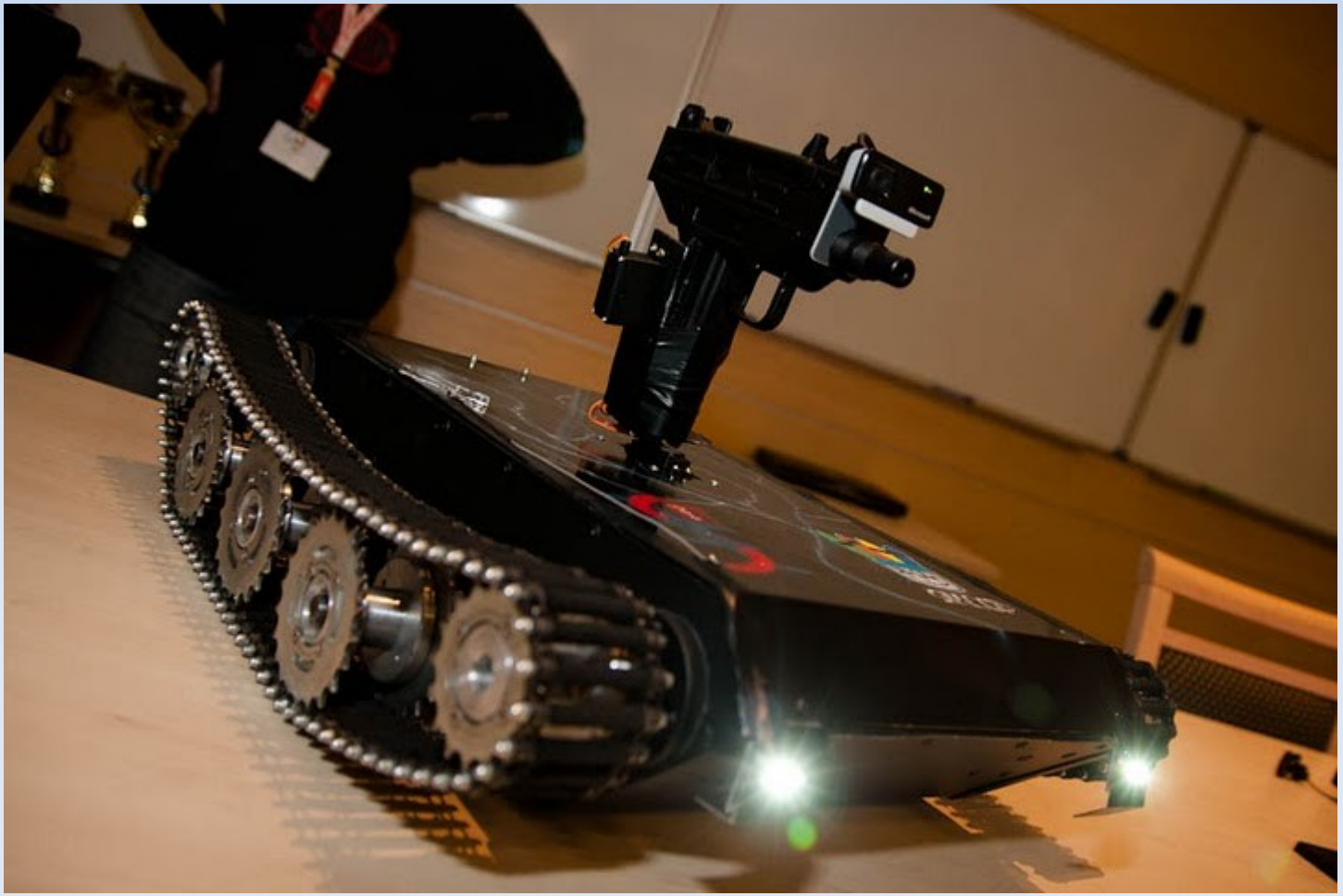
Konrad Brodzik

Plan prezentacji

- Co można zrobić z mikrokontrolerem?
- Czego do tego potrzeba?
- Jak to zrobić:
  - Język C na mikrokontrolery
  - Narzędzia konsolowe
  - IDE

Co można zrobić z mikrokontrolerem?

- Prezentacja sprzętu i filmików



Co można zrobić z mikrokontrolerem?

- Sterowanie
  - silniki (krokowe, serwo, szczotkowe, 3-fazowe)
  - wyświetlacze LCD
  - oświetlenie
- Komunikacja
  - USART (RS232, Bluetooth, IrDA)
  - USB
  - I^2C
  - Ethernet (uIP + >= Atmega128)

Co można zrobić z mikrokontrolerem?

- Analog
  - konwerter ADC (vumeter, woltomierz)
- Wejście cyfrowe
  - próbkowane
  - wyzwalane zboczem
- .. tylko tyle? ;-) Aż tyle!

Czego do tego potrzeba?

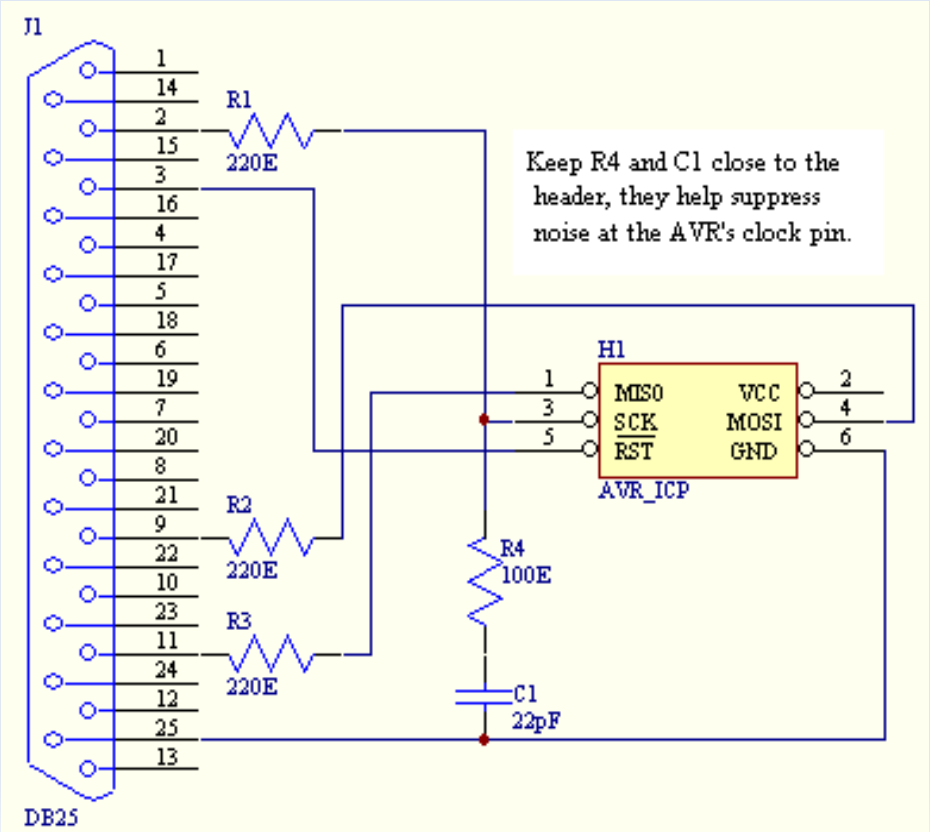
- mikrokontrolera
  - Od Attiny13 za 5 zł po Atmega644 (~ 30zł)
  - różnią się ilością wyprowadzeń (6-32) i pamięcią (4K-64K)

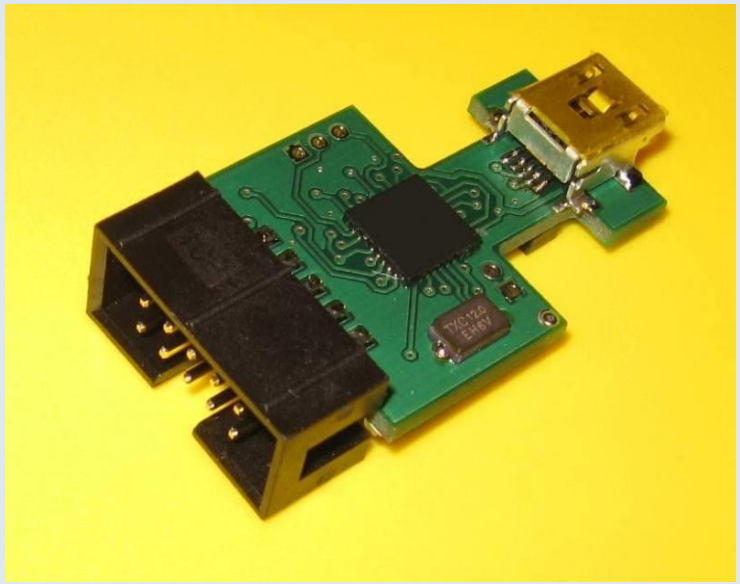
Czego do tego potrzeba?

- zasilania
  - Jakiekolwiek źródło 5V (baterie, zasilacz AT(X))

Czego do tego potrzeba?

- programatora
  - LPT za pół darmo
  - USB (STK500) – 30-50 zł





Czego do tego potrzeba?

- Opcjonalnie – zestaw prototypowy z wyświetlaczem, diodami, portem RS i stabilizatorem zasilania (80 zł za DIY, 300 zł zmontowany)

Język C na mikrokontrolery

Hello World czyli miganie diodą :)

Język C na mikrokontrolery

```
#define F_CPU 8000000UL


#include <util/delay.h>
#include <avr/io.h>


int main ()
{
	DDRC = _BV (PC0);		/* ustaw PC0 jako wyjscie */

	while (1)			/* nieskonczona petla */
	{

		PORTC &= ~_BV(PC0);
		_delay_ms(500);

		PORTC |= _BV(PC0);
		_delay_ms(500);
	}
	return 0;
}
```

Język C na mikrokontrolery

- Wpisywanie do rejestrów – 8 lub 16 bitowych
- Reset po zakończeniu programu głównego
- Obsługa przerwań specjalnymi funkcjami
- Użycie portów – cyfrowe wejście i wyjście – charakterystyczne dla konkretnego procesora!

Język C na mikrokontrolery

- Procyon AVRLIB
  - ładny, funkcjonalny kod
  - obsługa zaawansowanych funkcji (stos TCP, system plików FAT)
  - kilkukrotnie większe pliki binarne (dla servotest: 4,7 KB wobec 0,6B dla natywnego kodu)

Narzędzia konsolowe

- avr-gcc
- avrdude – obsługa programatora
- make
  - Standardowy Makefile http://electrons.psychogenic.com/modules/arms/art/8/AVRProjectOrganizationStandardizedAVRMakefileTemplate.php
- avra – asembler (składnia AT&T)
- simulavr - symulator (powolny!)

IDE

- Eclipse
  - konfigurowanie parametrów procesora
  - podpowiedzi (czego nie ma nawet oryginalne AVR Studio)
- Arduino IDE
  - ładne, ale..
  - mało konfigurowalne
  - drogi hardware

Pytania?