

System Programming

AVR-8 I/O Exercise

Hugo Marcondes

`hugom@lisha.ufsc.br`

`http://www.lisha.ufsc.br/~hugom`

Sep 2006

Application

- Traditional Led Application
- Pseudo Code:

```
int main() {  
    btns_pressed = read_btn();  
    write_leds(btns_pressed);  
}
```

Hardware

- STK 500 Development Kit
- AVR ATmega16L
- 4 GPIO ports
 - PORTB -> Connected to the LEDs
 - PORTD -> Connected to the Buttons
- Each GPIO has three registers for control
 - Data – The data latch register for the port (**PORTx**)
 - Data Direction – Configure the direction of the port (**DDRx**)
 - Pin Data – Access the physical value on the port pin (**PINx**)

Configuration of the Ports

- **LEDS – Output**
 - Configure DDR as 0xFF
 - Write on Data Register
 - One to turn off
 - Zero to light on

- **Buttons – Input**
 - Configure DDR as 0x00
 - Read the Physical Value of the PINs

AVR-8 Tools

- All the tools preinstalled on `/usr/local/avr`
- **GNU GCC** AVR-8 Cross Compiler
- **avrdude** for Flash Programming

- Makefile provided:
 - `make main` -> Compile `main.c`
 - `make send` -> Send the main executable to platform
 - `make` -> Compile `main.c` and send to platform