

Projeto Final - Sistemas Baseados em Microprocessadores

Utilização de diversos sensores com microcontroladores AVR

João CORREIA¹, Nuno FREITAS²

¹ FEUP, up201704897@fe.up.pt

² FEUP, up201704905@fe.up.pt

Resumo: No âmbito da cadeira de SBMI, realizámos um projeto de forma a aprofundar os nossos conhecimentos a programar com microcontroladores AVR, como o ATmega328p.

Palavras-chave: AVR-GCC, ATmega328p, Microcontroladores, Arduino

1. Introdução

Neste projeto vamos utilizar diversos tipos de sensores e atuadores, controlando-os através de um programa que transferimos para o microcontrolador do Arduino, o ATmega328p. Vamos simular um sistema de iluminação, um sistema de entrada e saída para um parque de estacionamento e também um sistema de ventilação.

2. Materiais

- 1 sensor de temperatura
- 1 motor DC 3V-6V (para a ventoinha caso seja atingido um limite de temperatura)
- 1 sensor RFID
- 1 motor servo (simula uma cancela de um parque de estacionamento)
- 1 led verde (lugares de estacionamento livres)
- 1 led vermelho (sem lugares de estacionamento livres)
- 1 sensor de luminosidade
- 1 led branco (liga caso fique demasiado escuro)
- 1 piezzo buzzer (sinaliza quando um cartão RFID é lido)
- 1 botão (opção extra de pressionar o botão para diminuir o número de lugares ocupados atualmente)
- 1 Arduino
- Fios, Resistências, Díodo, Transistor

3. Métodos e Procedimentos

3.1 Sensor de Temperatura

O sensor de temperatura utilizado foi o ds18b20. Este utiliza o OneWire como protocolo de comunicação. É enviado um comando de inicialização (reset pulse) pelo ATmega328p. Quando há confirmação da parte do sensor, é enviado um comando Skip Rom [CCh] que comunica com todos os dispositivos que utilizem OneWire (neste caso, o sensor de temperatura é o único), e depois é enviado um comando [44h] que pede a conversão de temperatura. O microprocessador recebe então a informação em 16 bits (sendo os 4 primeiros zeros, ou seja, 12 bits de precisão). O pino do Arduino do led branco é depois facilmente ativado quando um limite de temperatura é ultrapassado.

3.2 Sensor RFID

O sensor RFID utilizado foi o RFI0-RC522. Este sensor utiliza o chip MFRC522, que faz uma comunicação sem fios a 13,56 MHz, baseada no protocolo MIFARE. Este chip tem um FIFO Buffer de 64 bits que consegue receber e enviar informação para o Arduino através de diversos protocolos. Para este projeto selecionou-se a comunicação através do protocolo SPI, já que foi o mesmo protocolo utilizado para o sensor de luminosidade. Neste protocolo não se pode controlar os dois “slaves” ao mesmo tempo por isso foi necessário ir alterando o slave select (ou chip select) quando se queria comunicar com um ou outro dispositivo.

3.3 Sensor de Luminosidade

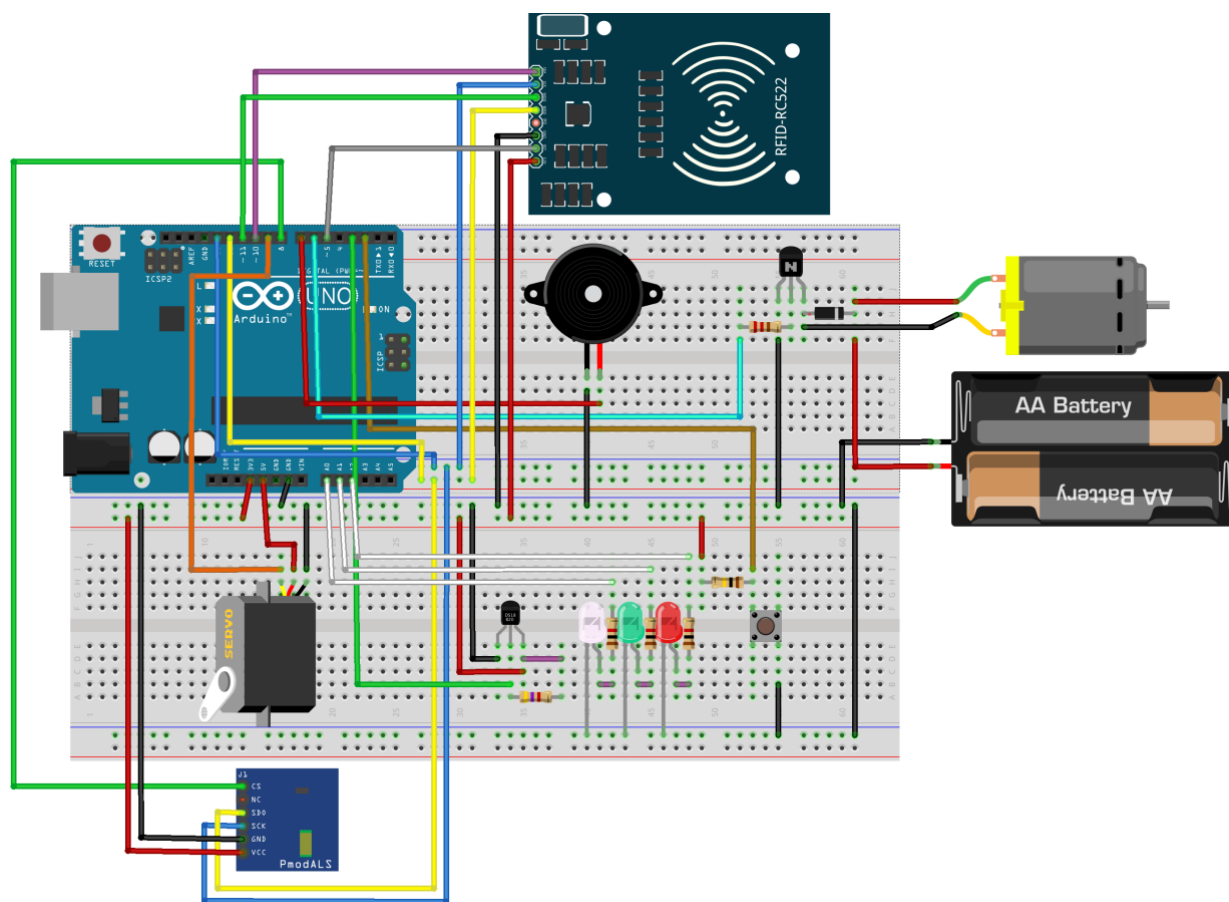
O sensor de luminosidade utilizado foi o PmodALS. Este comunica através do protocolo SPI. Quando a linha CS (chip select) ou SS (slave select) fica a zero, o microprocessador comunica com o PmodALS. Logo de seguida, é feita uma única leitura durante os próximos 16 ciclos do pino SCK (serial clock). Este tem de estar entre 1MHz e 4MHz para o PmodALS conseguir comunicar sem erros. No nosso caso utilizámos um divisor de 16 para diminuir a frequência da porta SCK de 16MHz (intrínseca ao Arduino Uno) para 1MHz. Os bits com a informação são colocados no “falling edge” do SCK e válidos no “rising edge” seguinte do SCK. Esta informação consiste de três zeros, oito bits com a luminosidade (com o bit mais significativo primeiro) e quatro zeros. Ficamos assim com um valor de luminosidade de oito bits (0-255) que nos permite facilmente detetar quando está de noite ou de dia.

3.4 Motor Servo

O motor servo é controlado através de PWM. Este modelo pode ser colocado entre 0 graus e 180 graus. Ele funciona a uma frequência de 50Hz (período de 20ms). O pulso PWM varia entre 1 ms e 2 ms para 0 graus e 180 graus respetivamente.

Toda a informação sobre como utilizar PWM com o servo está comentado no código e é complementado com a informação da datasheet do ATmega328p.

4. Circuito



fritzing

5. Funcionamento Geral

Quando fazemos reset ao Arduino, para além de algumas inicializações (spi, serial port, etc), existe um período de tempo, até o botão ser pressionado, em que novos cartões podem ser adicionados à base de dados. Depois de serem adicionados todos os cartões desejados, clicamos no botão para entrar no ciclo principal do programa. A partir deste momento, apenas os cartões adicionados são aceites pelo leitor para mover o servo 90 graus, simulando um portão. Se existirem lugares disponíveis o led verde estará ligado e caso contrário, se todos os lugares estiverem ocupados, o led vermelho estará ligado. O led branco é aceso caso a luminosidade seja mais baixa que um determinado valor que podemos definir no programa. Temos ao mesmo tempo uma ventoinha controlada por um simples motor DC que é ativada quando um limite de temperatura é ultrapassado, que também pode ser definido no programa.

FIM do Relatório