

Mikroprosessorsystemer

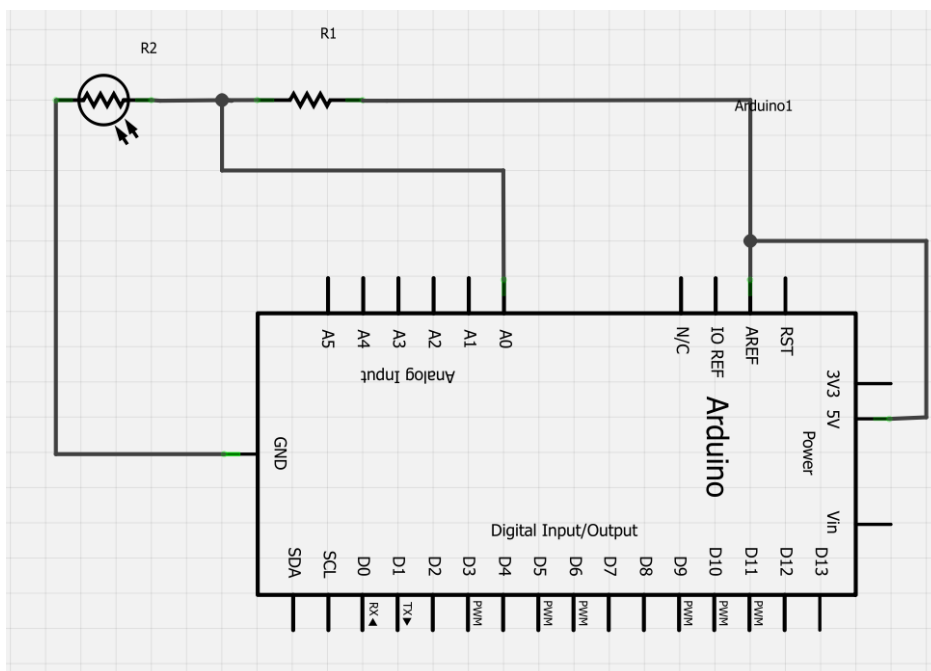
Labøving 9 –Analog til digitalkonverter (ADC).

Oppgave 1.

Bruk Arduino UNO og lag et program som konfigurerer ADC til å sample en inngang og sende ut resultatet på uart. Dere kan enten bruke egne rutiner, rutinene til Elliot Williams eller bruke printf i avr-libc.

For å bruke printf må se kodeeksempel på neste side

Koble opp en lysfølsom motstand (LDR) til og mål denne. Forslag til koblingskjema:



Vi kan bruke AVCC som referanse for koblingen over. R1 kan være 10k Ω . Datablad: [Lyssensor](#)

Oppgave 2.

Bruke temperatursensoren dere har i Arduinosettet deres og skriv ut temperaturen på UARTen.

Lever oversiktlig og kommentert c-kode i It's Learning.

Kodeeksempel for uart:

```
#define F_CPU 16000000UL
#define USART_BAUDRATE 9600 // desired baud rate
#define UBRR_VALUE (((F_CPU / (USART_BAUDRATE * 16UL))) - 1) // UBRR value

/*****|Includes|*****/
#include <avr/interrupt.h>
#include <stdio.h>
/*****|function prototypes|*****/
static int usart_putchar( char data, FILE *stream );
void USART_init(uint16_t ubrr_value);
/*****|

static FILE uart_str = FDEV_SETUP_STREAM(usart_putchar, NULL, _FDEV_SETUP_WRITE); // for printf to work

int main(void)
{
    USART_init(UBRR_VALUE);

    DDRC = 0x00;
    // port c is now input, pinc0 is adc0

    stdout = &uart_str;
    // to make printf work

    while (1)
    {
        printf("ADC data = %d\r",1000);
        // prints out ADC value to USART
        //_delay_ms(250);
    }
}

/*****|
static int usart_putchar(char data, FILE *stream) {
    while ((UCSR0A & (1 << UDRE0)) == 0) {};
    // Wait for empty transmit buffer
    UDR0 = data; // Start transmission
    return 0;
}

/*****|usart init|*****/
void USART_init(uint16_t ubrr_value)
{
    UBRR0 = ubrr_value;
    // set baud rate to 9600

    UCSR0C = ((1<<UMSEL01)|(3<<UCSZ00));
    // 8 bit 1 parity
    UCSR0B = ((1<<TXEN0)|(1<<RXEN0));
    // enable transmitter, reciever and interrupt
}
```