

Laborator 13

ADC

Obiective

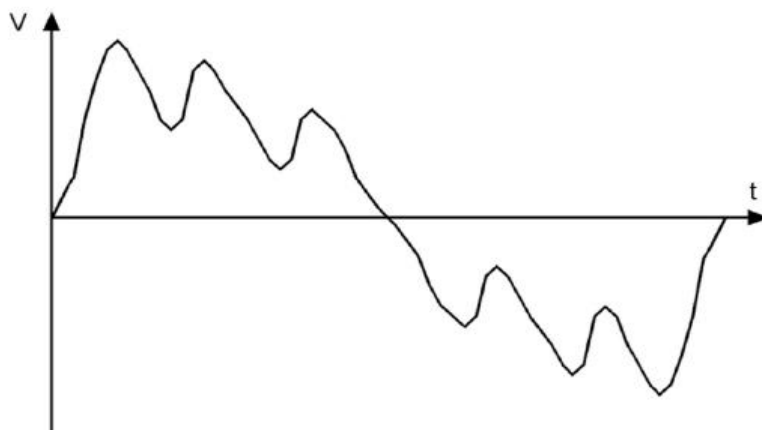
- Înțelegerea modului de funcționare a convertorului analog-digital (ADC)

Cuprins

Obiective.....	1
Cuprins.....	1
Noțiuni introductive	2
ADC la ESP8266.....	5
Fotorezistorul	6
Aplicații	7
Scopul laboratorului	7

Noțiuni introductive

Lumea înconjurătoare este alcătuită în esență din semnale analogice: lumina, sunetul, căldura. Pentru a putea fi măsurate într-un sistem de calcul digital, aceste semnale trebuie convertite în valori numerice discrete.

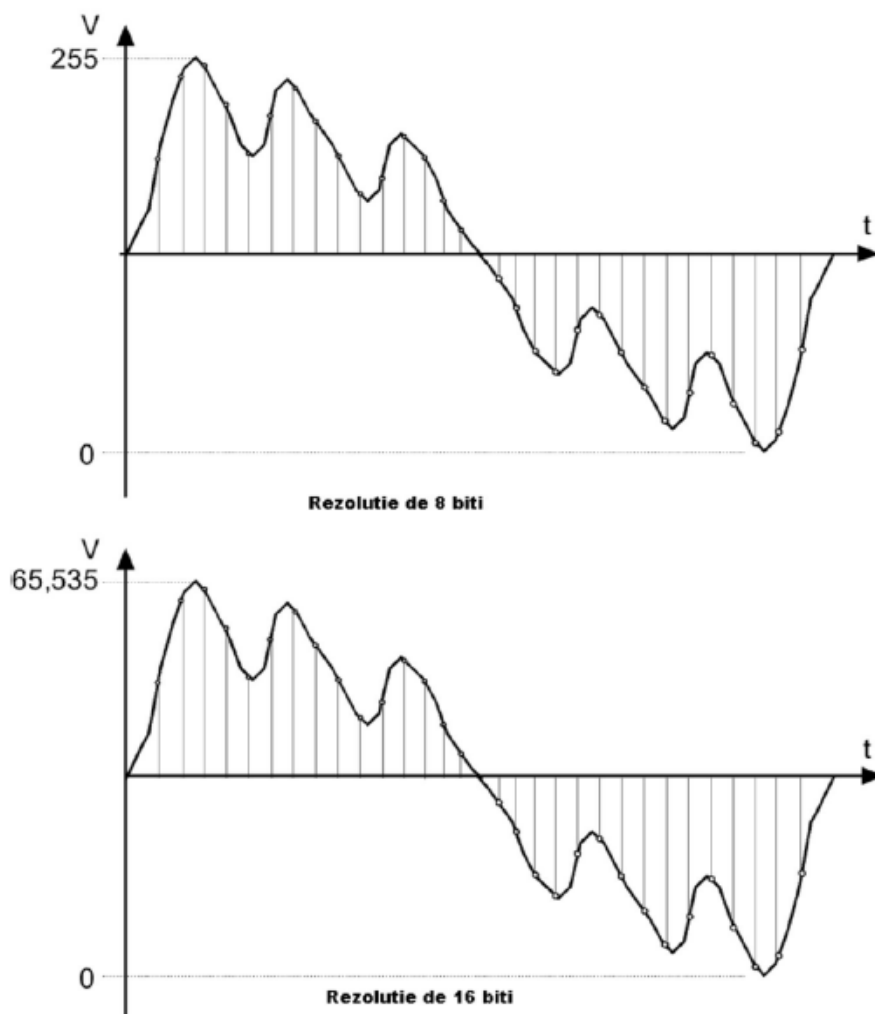


Figură 1 Reprezentarea unui semnal analogic

Un convertor analog – digital (ADC) este un circuit electronic care convertește o tensiune analogică de la intrare într-o valoare digitală. Aceasta poate fi reprezentată în mai multe feluri în funcție de codificarea datelor : binar, cod Gray sau cod complement al lui doi.

O caracteristică importantă a unui ADC o constituie rezoluția acestuia. Rezoluția indică numărul de valori discrete pe care convertorul poate să le furnizeze la ieșirea sa în intervalul de măsură. Deoarece rezultatele conversiei sunt de obicei stocate intern sub formă binară, rezoluția unui convertor analog-digital este exprimată în biți.

De exemplu, dacă **rezoluția** unui convertor este de 10 biți atunci el poate furniza $2^{10} = 1024$ valori diferite la ieșire. Dacă gama de măsurare este de 0-5V, rezoluția de măsurare va fi : $(5V - 0V) / 1024 = 0,005V = 5mV$.



Figură 2 Acelasi semnal analogic esantionat la rezolutii diferite

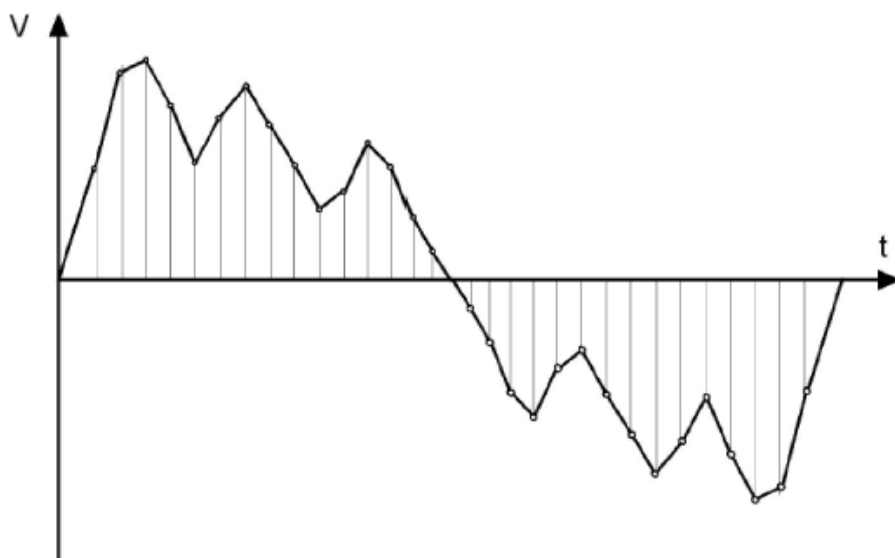
O altă caracteristică importată a unui convertor analog-digital o constituie **rata de eșantionare**.

Această depinde de timpul dintre două conversii succesive și afectează modul în care forma de undă originală va fi redată după procesarea digitală.

Practic, se vor cunoaște valori din semnalul original doar în anumite momente de timp corespunzătoare momentului conversiei, deci reconstituirea semnalului nu va putea reproduce perfect originalul.

Figura 4 arată modul în care semnalul eșantionat din figura de mai sus va fi reconstituit în urmă trecerii printr-un convertor digital – analog (DAC). După cum se poate observa, semnalul reprodus nu este identic cu cel original.

Dacă rata de eșantionare ar crește, este de la sine înțeles că semnalul reprodus ar aproxima din ce în ce mai bine originalul.



Figură 4 Semnalul analogic refacut în urma conversiei inverse (DAC)

Care este însă rată minimă de eșantionare pentru a reproduce fără pierderi un semnal de o frecvență dată?

Teorema lui Nyquist spune că o rată de eșantionare de minim două ori mai mare decât frecvența semnalului măsurat este necesară.

Acest lucru poate fi aplicat și dacă avem în schimb un semnal compus dintr-o mulțime de frecvențe, cum ar fi vocea umană sau o melodie.

Limitele maxime ale auzului uman sunt 20Hz – 20kHz dar frecvențele obișnuite pentru voce sunt în gama 20-4000Hz, de aceea centralele telefonice folosesc o rată de eșantionare a semnalului de 8000Hz.

Rezultatul este o reproducere inteligibilă a vocii umane, suficientă pentru transmiterea de informații într-o convorbire obișnuită.

Pentru reproducerea fidelă a spectrului audibil se recurge la rate mai mari de eșantionare. De exemplu, înregistrarea pe un CD are o rată de eșantionare de 44100Hz ceea ce este mai mult decât suficient pentru reproducerea fidelă a tuturor frecvențelor audibile.

În funcție de modul în care se execută conversia, convertoarele analog-digitale pot fi de mai multe tipuri:

- ADC paralel (Flash)
- ADC cu aproximare succesivă
- ADC cu integrare (single-slope, dual-slope);
- ADC Sigma-delta (delta-sigma, 1-bit ADC sau ADC cu oversampling).

ADC la ESP8266

ADC-ul de pe microcontrolerul **ESP8266** este un ADC pe 10 biți. Intrarea convertorului se găsește pe **pinul A0** (pe unele plăci de dezvoltare se mai găsește sub denumirile TOUT, Analog Pin 0, ADC, Pin6).

Convertorul analog digital poate citi tensiuni cu valori cuprinse între 0.00V – 3.3V, așadar pentru o tensiune de intrare de 0V ADC-ul va returna valoarea 0, iar pentru o tensiune de intrare de 3.3V ADC-ul va returna valoarea 1023. (deci avem o precizie de 3.2mV).

ADC-ul se configureaza cu ajutorul următoarelor structuri de date și funcții:

```
typedef struct {  
    adc_mode_t mode;  /*!< ADC mode */  
    uint8_t clk_div;  /*!< ADC sample collection clock=80M/clk_div, range[8, 32] */  
} adc_config_t;  
  
typedef enum {  
    ADC_READ_TOUT_MODE = 0,  
    ADC_READ_VDD_MODE,  
    ADC_READ_MAX_MODE  
} adc_mode_t;
```

Primul element al structurii este o enumerație și descrie modul în care se va configura ADC-ul. Al doilea element al structurii este un element întreg fara semn pe 8 biți și reprezintă factorul de divizare a frecvenței de clock (80Mhz) pentru a obține frecvența de eșantionare, intervalul de valori valide pe care le lua acest factor este [8, 32].

Funcția `esp_err_t adc_init(adc_config_t *config);` este funcția de inițializare a ADC-ului și ia ca parametru structura de date descrisă mai sus.

Funcția poate returna:

- `ESP_OK Success`
- `ESP_ERR_NO_MEM malloc fail`
- `ESP_ERR_INVALID_ARG Parameter error`
- `ESP_FAIL adc has been initialized`

Funcțiile pentru citire sunt:

- `esp_err_t adc_read(uint16_t *data);`
- `esp_err_t adc_read_fast(uint16_t *data, uint16_t len);`

Mai există și o funcție de dezinițializare `esp_err_t adc_deinit();`

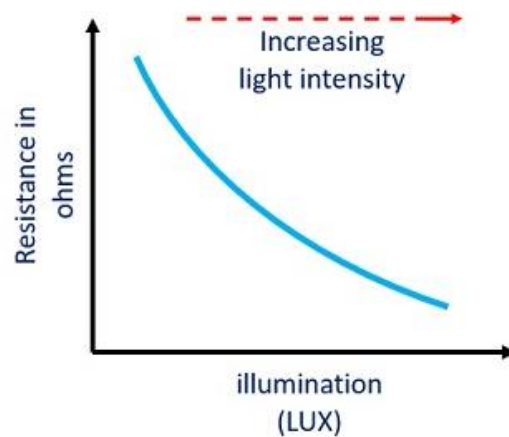
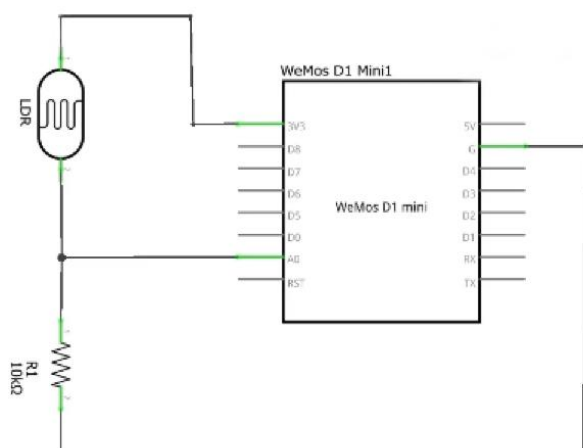
Fotorezistorul



Fotorezistorul este un element pasiv de circuit căruia îi variază rezistența în funcție de intensitatea luminii ce ajunge pe el.

În general, fotorezistențele sunt de putere mică și au rezistențe începând de la câțiva $k\Omega$ la lumină până la rezistențe de ordinul $M\Omega$ -lor în întuneric.

Cea mai simplă aplicație este un circuit cu un divizor rezistiv format din fotorezistență și o altă rezistență de valori mari. Prin variația rezistenței de la fotorezistență, variază și tensiunea de ieșire.



Aplicații

Funcție	Parametrii	Descriere
PHRES_u16Read(void)	-	Funcția returnează valoarea citită de ADC din momentul curent.
PHRES_vTaskCalculate(void)	-	Funcția actualizează valoarea din structura de date corespunzătoare fotorezistorului cu valoarea returnată de funcția PHRES_u16Read(void)

Scopul laboratorului

1. Creați funcțiile PHRES_u16Read și PHRES_vTaskCalculate.
2. Afișați nivelul de luminozitate în aplicația web, în chenarul de fotorezistor.
3. *În funcție de valoarea citită de pe fotorezistor, aprindeți led-ul RGB.