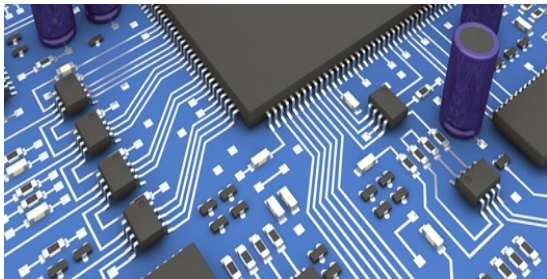


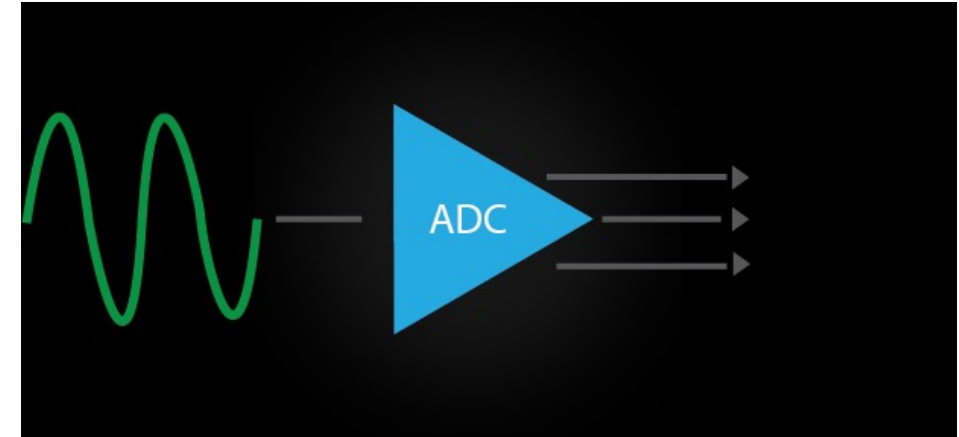
Aplikace Embedded systémů v Mechatronice



Michal Bastl

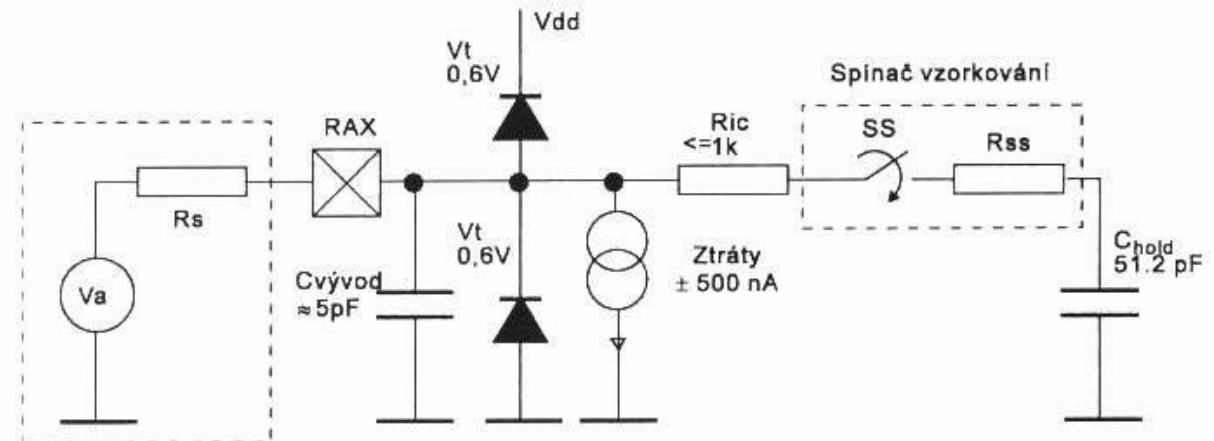
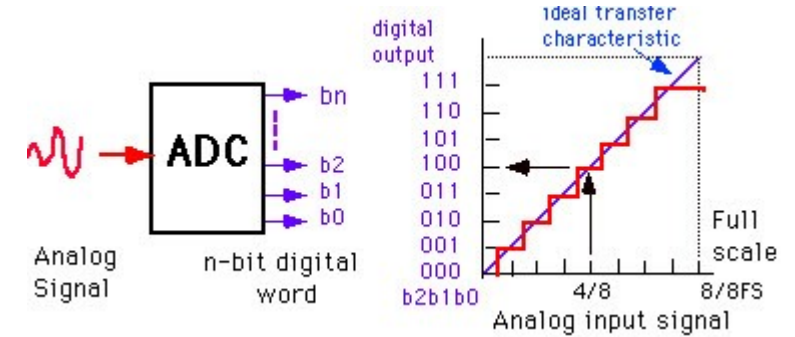
Analogově digitální převod

- Úkolem AD převodníku je převést analogovou (spojitou) hodnotu napětí na pinu na její digitální reprezentaci.
- Takových převodníků existuje celá řada, my se zaměříme na převodníky v MCU a to je nejčastěji převodník SAR.
- SAR znamená převodník s postupnou aproximací.
- Převodník provede konverzi analogové hodnoty na její binární reprezentaci.
- Důležité je tedy vždy počítat s tím, že výsledek takového převodu je přibližná hodnota.



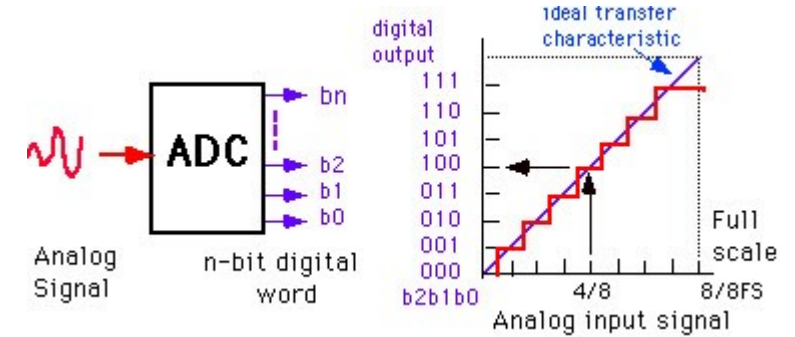
Analogově digitální převod

- Analogově digitální převodník převádí analogovou hodnotu napětí na její digitální interpretaci.
- Proto jedním ze základních parametrů je rozlišení převodníku.
- Jedná se o počet hladin na kterou dokáže převodník rozdělit interval referenčního napětí.
- ADC převodník na PIC 18 pracuje na principu postupné aproximace (Successive approximation ADC).

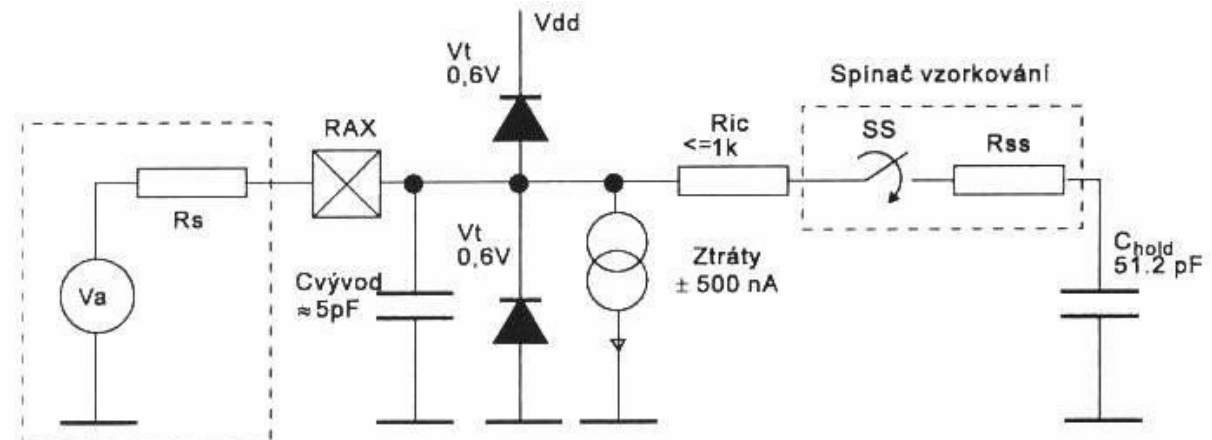


Analogově digitální převod

- AD převodník tedy nemá lineární převodní charakteristiku (obr. Vpravo) je schodovitá
- AD převodník obsahuje sample-and-hold SHA část, která připojí pin k internímu kondenzátoru, ten se nabije na hodnotu napětí na pinu.
- Musím tedy připojit kondenzátor na nějakou dobu, aby byla hodnota spolehlivě správná (setrvačný děj)
- Je třeba počítat s tím, že potřebuji i určitý elektrický náboj, který se do kondenzátoru přesune. Blízko pinu se dává ještě externí kondenzátor.
- Dnešní ADC v MCU mají rychlosti v řádu kHz i MHz



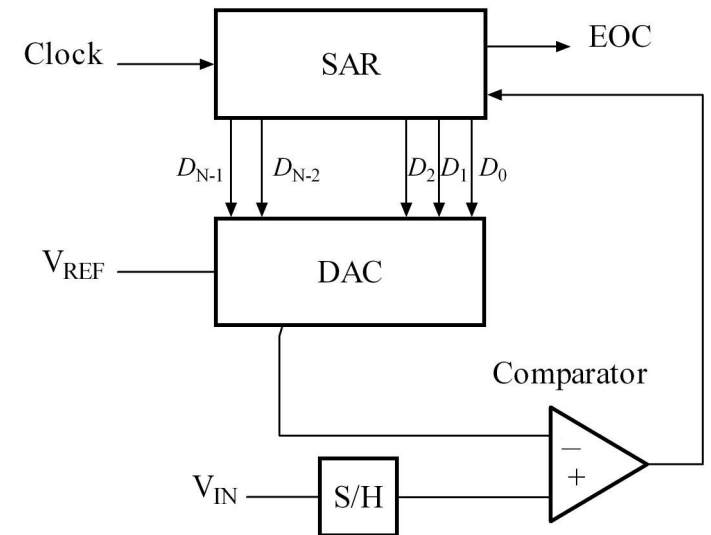
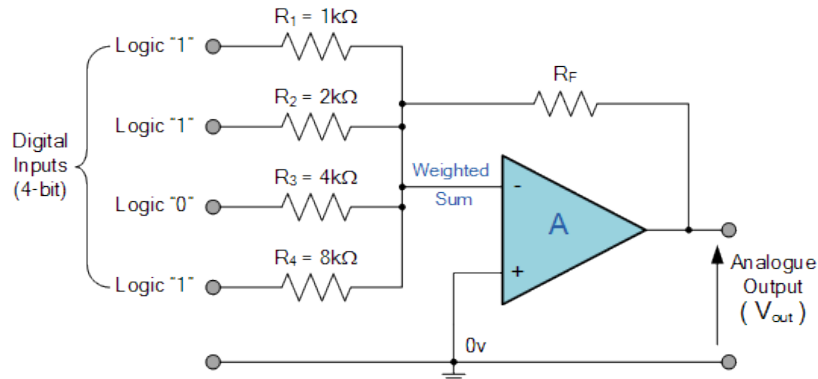
Přibližné elektrické schéma SHA obvodu v PIC18



ADC převodník PIC

- ADC převodník MCU PIC18 se skládá z DAC převodníku, který slouží k nastavení požadované úrovně napětí
- Komparátorem je rozhodnuto, jestli napětí na AN pinu je větší nebo menší než nastavená hodnota
- Takto se projde všech 10 bitů
- Jedna se o analogii půlení intervalu

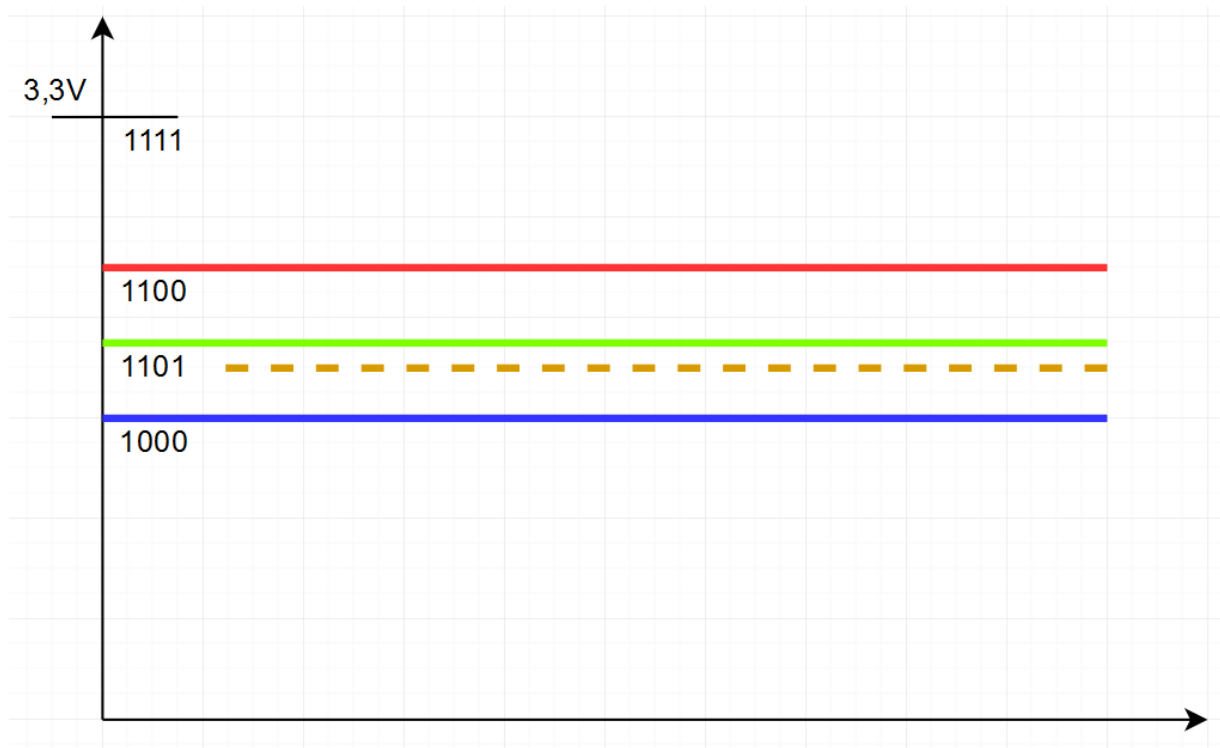
Zjednodušený DAC převodník:



$$U_{in} = ADC \cdot \frac{U_{ref}}{ADC_{res}}$$

ADC převodník PIC

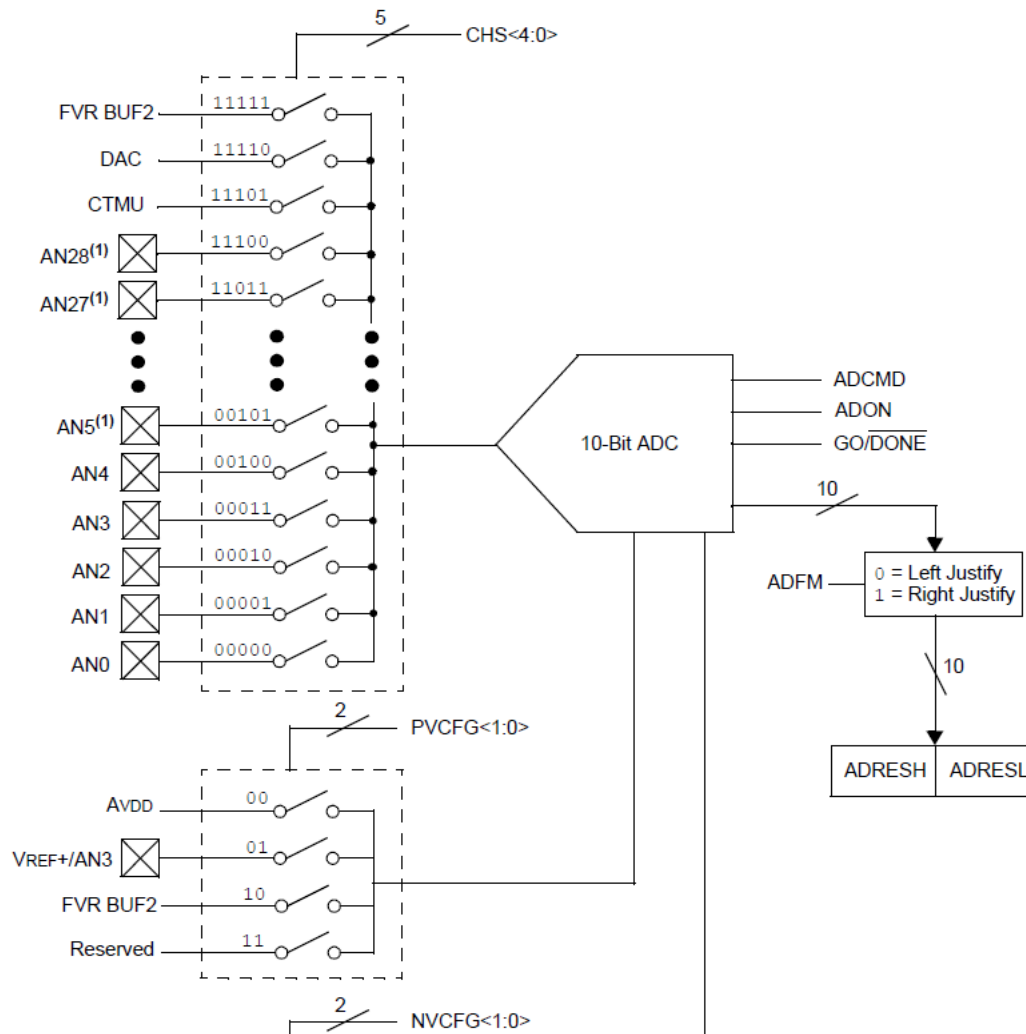
- Převod probíhá od nejvyššího bitu ten vždy binárně rozdělí interval na polovinu.



- Nejvyšší bit vždy rozdělí hodnotu na polovinu
- Každý další rozdělí předchozí polovinu zase na polovinu
- Tak to jde přes všechny bity
- Tímto se přiblížím ke skutečné hodnotě
- Proto je podstatné rozlišení převodníku, které se udává v počtu bitů
- Na PIC18 je k dispozici 10-bit převodník

ADC převodník PIC18

FIGURE 17-1: ADC BLOCK DIAGRAM



- Převodník je v MCU je v našem MCU jeden
- Pokud chci měřit napětí na více pinech, musím přepínat kanály
- To se dělá v sekci ADCON0bits.CHS
- Dále lze převodníku volit napětové reference pro kladnou a zápornou hodnotu
- Na RevKitu nastavujeme jako referenci napájecí napětí MCU, což je jednoduché, ale pro přesnější měření to není vhodné
- Pro přesnější měření se používají přesné napětové reference, které mohou přivést například na piny Vref

ADC převodník PIC18

Pokud chceme používat AD převodník integrovaný v mikrokontroléru PIC18F46K22, musíme vykonat následující kroky:

- Nastavení PORTů
- Volba kanálu
- Volba zdroje referenčního napětí (defaultní hodnota je napětí MCU na pinech AVDD a AVSS)
- Volba časování převodu
- Nastavení přerušení (pokud chceme)
- Formát výsledku převodu

ADC převodník PIC18

Perioda převodu:

- časování AD převodu se volí bity ADCS (ADCON2).
- Doba převodu jednoho bitu je definována jako TAD.
- Kompletní A/D převod vyžaduje 11TAD.
- FOSC/2
- FOSC/4
- FOSC/8
- FOSC/16
- FOSC/32
- FOSC/64
- FRC(interní RC oscilátor)
- Je třeba zvolit správně TAD min. 1 μ s.
- Mohu volit také dobu po kterou je připojen (nabíjen) samplovací kondenzátor
- Delší čas nabíjení kondenzátoru znamená spolehlivější měření
- Doba připojení kondenzátoru se počítá na počet TAD
- Tedy počet period ADC po kterou je kontrolní kondenzátor připojen

ADC převodník PIC18

REGISTER 17-3: ADCON2: A/D CONTROL REGISTER 2

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	ACQT<2:0>			ADCS<2:0>		
bit 7							bit 0

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'
 -n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

bit 7 **ADFM:** A/D Conversion Result Format Select bit
 1 = Right justified
 0 = Left justified

bit 6 **Unimplemented:** Read as '0'

bit 5-3 **ACQT<2:0>:** A/D Acquisition time select bits. Acquisition time is the duration that the A/D charge holding capacitor remains connected to A/D channel from the instant the GO/DONE bit is set until conversions begins.

000 = 0⁽¹⁾
 001 = 2 TAD
 010 = 4 TAD
 011 = 6 TAD
 100 = 8 TAD
 101 = 12 TAD
 110 = 16 TAD
 111 = 20 TAD

bit 2-0 **ADCS<2:0>:** A/D Conversion Clock Select bits

000 = FOSC/2
 001 = FOSC/8
 010 = FOSC/32
 011 = FRC⁽¹⁾ (clock derived from a dedicated internal oscillator = 600 kHz nominal)
 100 = FOSC/4
 101 = FOSC/16
 110 = FOSC/64
 111 = FRC⁽¹⁾ (clock derived from a dedicated internal oscillator = 600 kHz nominal)

`ANSELA = 0b00100000;` //AN4

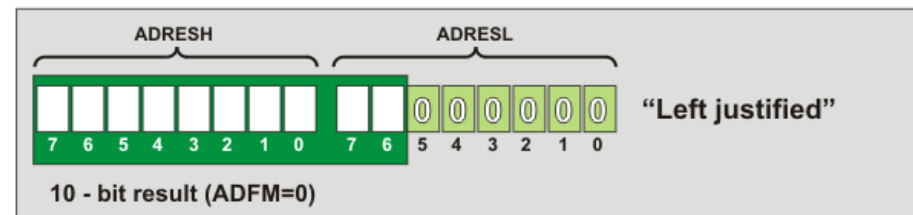
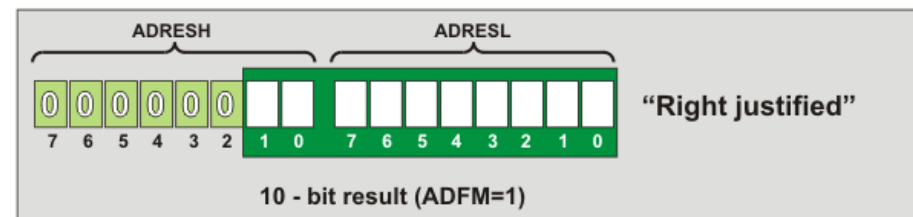
`ANSELE = 0b1;` //AN5

`ADCON2bits.ADFM = 1;` //right justified

`ADCON2bits.ADCS = 0b110;` //Fosc/64

`ADCON2bits.ACQT = 0b110;` //16 TAD

`ADCON0bits.ADON = 1;` //ADC turn on



ADC převodník PIC18

REGISTER 17-1: ADCON0: A/D CONTROL REGISTER 0

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	CHS<4:0>					GO/DONE	ADON
bit 7							bit 0

Legend:

R = Readable bit
-n = Value at POR

W = Writable bit
'1' = Bit is set

U = Unimplemented bit, read as '0'
'0' = Bit is cleared

x = Bit is unknown

bit 7 Unimplemented: Read as '0'

bit 6-2 CHS<4:0>: Analog Channel Select bits

00000 = AN0
00001 = AN1
00010 = AN2
00011 = AN3
00100 = AN4
00101 = AN5⁽¹⁾
00110 = AN6⁽¹⁾
00111 = AN7⁽¹⁾
01000 = AN8
01001 = AN9
01010 = AN10
01011 = AN11
01100 = AN12
01101 = AN13
01110 = AN14
01111 = AN15
10000 = AN16
10001 = AN17
10010 = AN18
10011 = AN19
10100 = AN20⁽¹⁾
10101 = AN21⁽¹⁾
10110 = AN22⁽¹⁾
10111 = AN23⁽¹⁾
11000 = AN24⁽¹⁾
11001 = AN25⁽¹⁾
11010 = AN26⁽¹⁾
11011 = AN27⁽¹⁾
11100 = Reserved
11101 = CTMU
11110 = DAC
11111 = FVR BUF2 (1.024V/2.048V/2.096V Volt Fixed Voltage Reference)⁽²⁾

bit 1 GO/DONE: A/D Conversion Status bit

1 = A/D conversion cycle in progress. Setting this bit starts an A/D conversion cycle.
This bit is automatically cleared by hardware when the A/D conversion has completed.
0 = A/D conversion completed/not in progress

bit 0 ADON: ADC Enable bit

1 = ADC is enabled
0 = ADC is disabled and consumes no operating current

ANSELA = 0b00100000; //AN4

ANSELE = 0b1; //AN5

ADCON2bits.ADFM = 1; //rightt justified

ADCON2bits.ADCS = 0b110; //Fosc/64

ADCON2bits.ACQT = 0b110; //16

ADCON0bits.ADON = 1; //ADC turn on

ADCON0bits.CHS = 5; //select AN5

GODONE = 1; //start

while(GODONE); //wait until its done

pot1 = (ADRESH << 8) | ADRESL; //combine registers

ADC převodník PIC18

REGISTER 17-2: ADCON1: A/D CONTROL REGISTER 1

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
TRIGSEL	—	—	—	PVCFG<1:0>		NVCFG<1:0>	
bit 7							bit 0

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'
-n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

bit 7 **TRIGSEL:** Special Trigger Select bit
1 = Selects the special trigger from CTMU
0 = Selects the special trigger from CCP5

bit 6-4 **Unimplemented:** Read as '0'

bit 3-2 **PVCFG<1:0>:** Positive Voltage Reference Configuration bits
00 = A/D VREF+ connected to internal signal, AVDD
01 = A/D VREF+ connected to external pin, VREF+
10 = A/D VREF+ connected to internal signal, FVR BUF2
11 = Reserved (by default, A/D VREF+ connected to internal signal, AVDD)

bit 1-0 **NVCFG<1:0>:** Negative Voltage Reference Configuration bits
00 = A/D VREF- connected to internal signal, AVss
01 = A/D VREF- connected to external pin, VREF-
10 = Reserved (by default, A/D VREF- connected to internal signal, AVss)
11 = Reserved (by default, A/D VREF- connected to internal signal, AVss)

- Trigsels znamená, že mohu spustit konverzi na nějakou jinou událost. Zájemce si může najít vysvětlení v datasheetu v našem předmětu se tím nebudeme zabývat.
- Je zde positive voltage reference a negative voltage reference
- My používáme prostě $-V_{ref} = 0V$
- $+V_{ref} = 3,3V$
- Mohl bych použít zdroj napětíové reference a přivést ho na příslušné piny

Využití vnitřní reference MCU

REGISTER 21-1: VREFCON0: FIXED VOLTAGE REFERENCE CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-1	U-0	U-0	U-0	U-0
FVREN	FVRST	FVRS<1:0>	—	—	—	—	—
bit 7							bit 0

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7	FVREN: Fixed Voltage Reference Enable bit 0 = Fixed Voltage Reference is disabled 1 = Fixed Voltage Reference is enabled
bit 6	FVRST: Fixed Voltage Reference Ready Flag bit 0 = Fixed Voltage Reference output is not ready or not enabled 1 = Fixed Voltage Reference output is ready for use
bit 5-4	FVRS<1:0>: Fixed Voltage Reference Selection bits 00 = Fixed Voltage Reference Peripheral output is off 01 = Fixed Voltage Reference Peripheral output is 1x (1.024V) 10 = Fixed Voltage Reference Peripheral output is 2x (2.048V) ⁽¹⁾ 11 = Fixed Voltage Reference Peripheral output is 4x (4.096V) ⁽¹⁾
bit 3-2	Reserved: Read as '0'. Maintain these bits clear.
bit 1-0	Unimplemented: Read as '0'.

```
VREFCON0bits.FVRS = 0b10;
VREFCON0bits.FVREN = 1;
while(!FVRST);
```

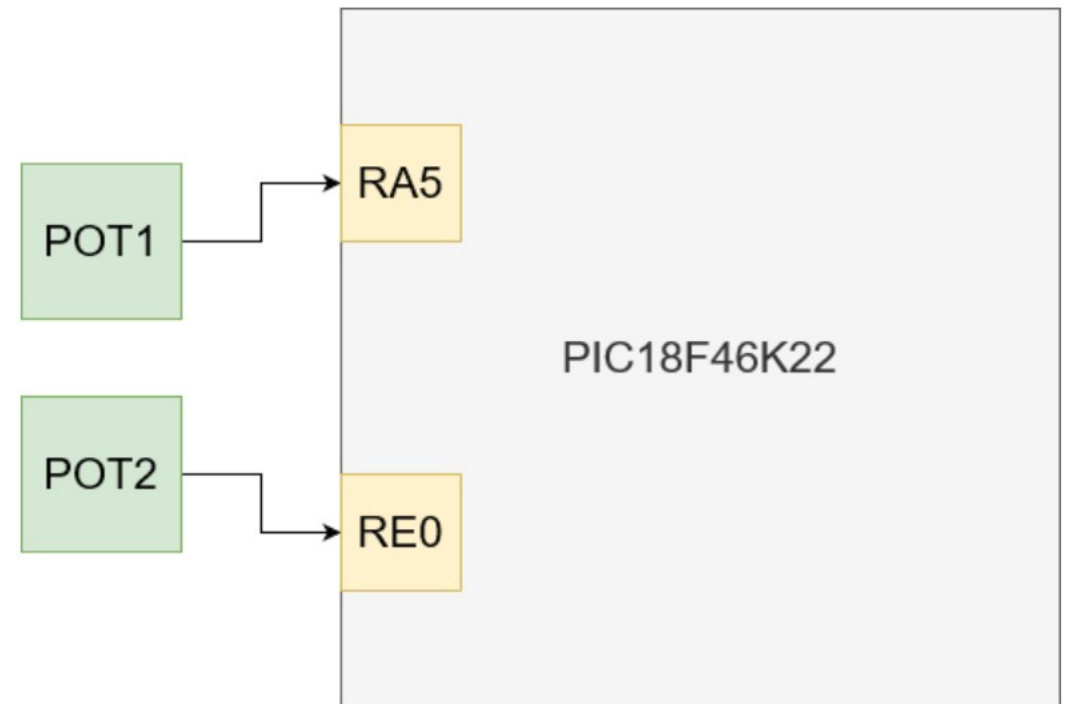
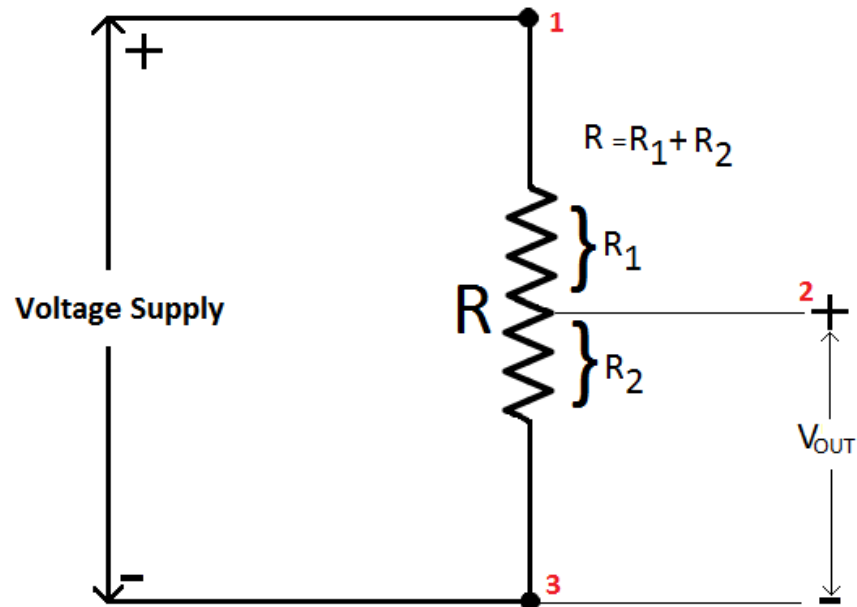
```
ADCON1bits.PVCFG = 0b10;
ADCON1bits.NVCFG = 0b00;
```

Param No.	Sym	Characteristics	Min	Typ	Max	Units	Comments
VR01	VROUT	VR voltage output to ADC	0.973	1.024	1.085	V	1x output, VDD ≥ 2.5V
			1.946	2.048	2.171	V	2x output, VDD ≥ 2.5V
			3.891	4.096	4.342	V	4x output, VDD ≥ 4.75V (PIC18F2X/4XK22)
VR02	VROUT	VR voltage output all other modules	0.942	1.024	1.096	V	1x output, VDD ≥ 2.5V
			1.884	2.048	2.191	V	2x output, VDD ≥ 2.5V
			3.768	4.096	4.383	V	4x output, VDD ≥ 4.75V (PIC18F2X/4XK22)
VR04*	TSTABLE	Settling Time	—	25	100	μs	0 to 125°C

* These parameters are characterized but not tested.

ADC převodník PIC18

- Na našem RevKitu máme dvojici potenciometrů
- Ty jsou připojeny na piny RA5 a RE0
- Tyto piny odpovídají ADC kanálům AN4 a AN5



Praktické poznámky

Nyquistův–Shannonův vzorkovací teorém:

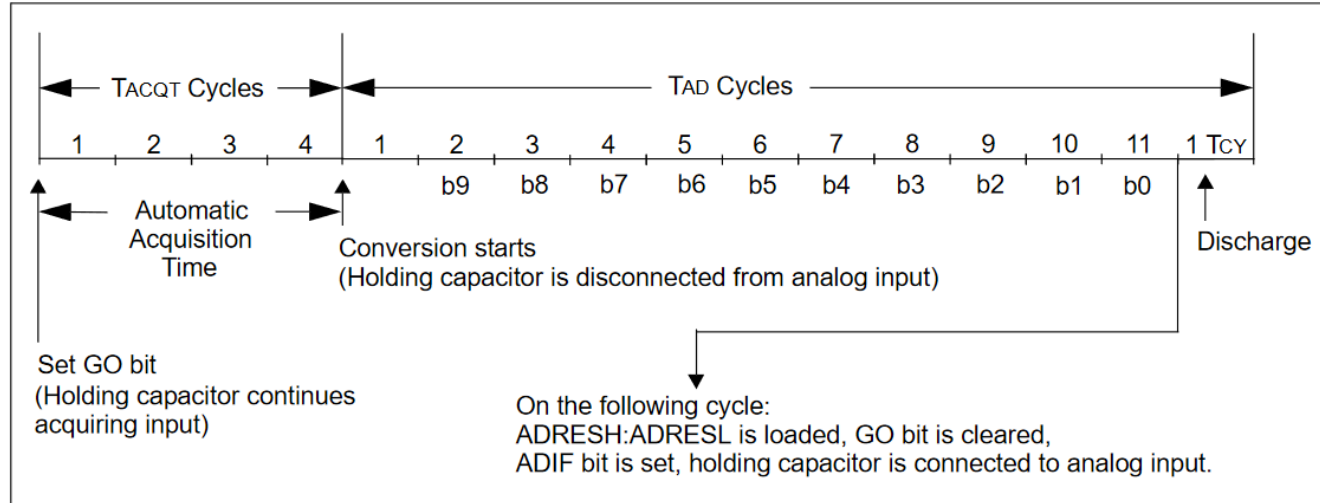
- „přesná rekonstrukce spojitého, frekvenčně omezeného signálu z jeho vzorků je možná tehdy, pokud byla vzorkovací frekvence vyšší než dvojnásobek nejvyšší harmonické složky vzorkovaného signálu.“
- Tvrzení platí pro rekonstrukci frekvence harmonického signálu.
- Při praktických měřeních a pro dobrou rekonstrukci amplitudy je zvykem volit 10x větší frekvenci vzorkování.

Aliasing:

- Problém, který vznikne se pak nazývá aliasing.
- Tomuto jevu zabráním přidáním anti-aliasingového filtru.
- Ten filtruje vyšší frekvence, v signálu, které nemohu vzorkováním zachytit.
- Tyto frekvence jsou v měřeném signálu často formou přirozeného šumu.

Praktické poznámky

FIGURE 17-4: A/D CONVERSION TAD CYCLES (ACQT<2:0> = 010, TACQ = 4 TAD)



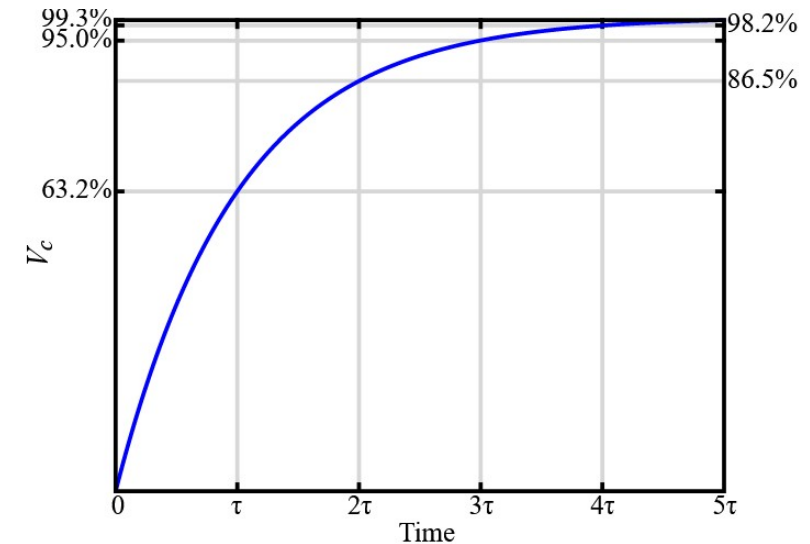
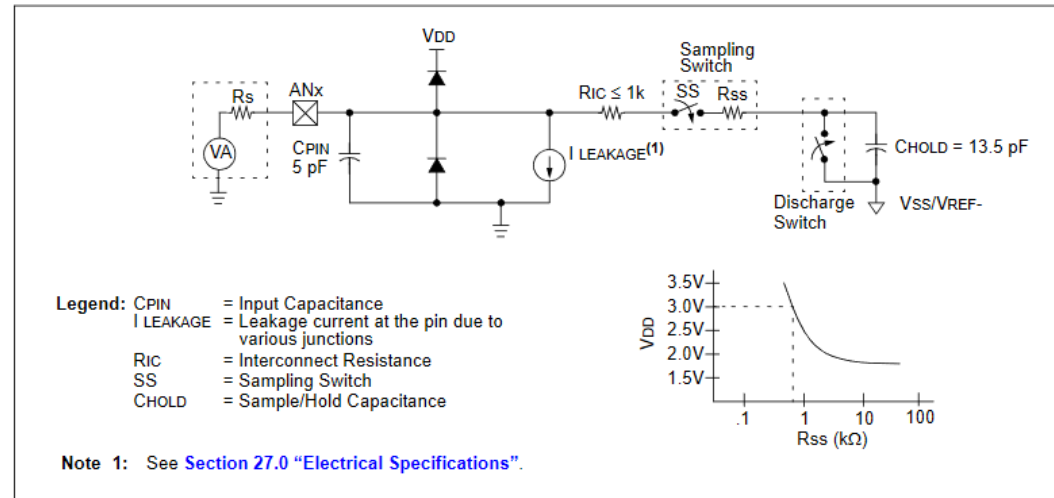
Rychle ověření:

$R_c < 10k$

$Chold = 13.5p$

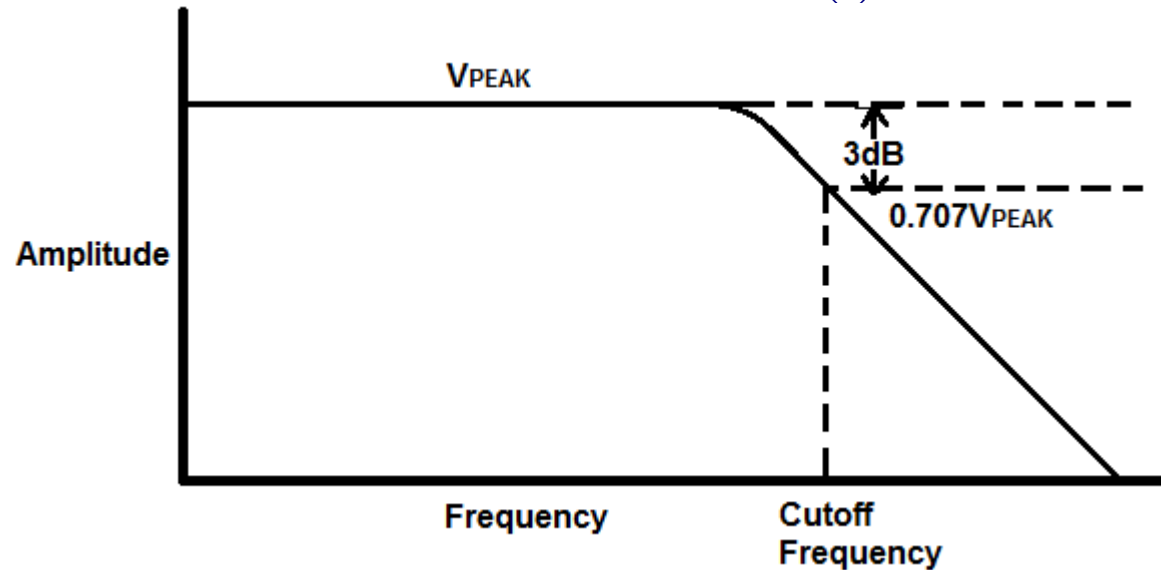
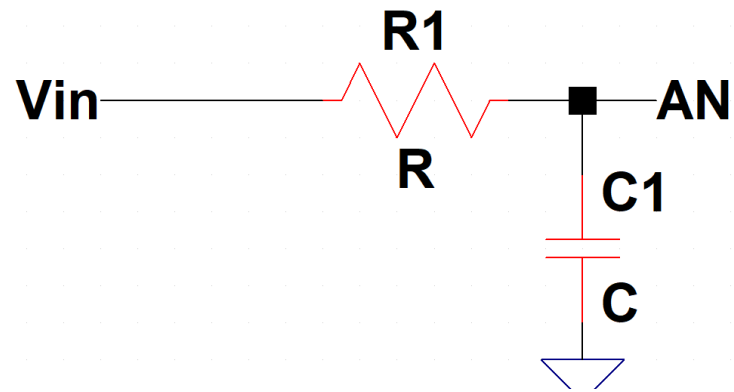
$F_c = 1\,178\,925\text{ Hz}$

FIGURE 17-5: ANALOG INPUT MODEL



Hardware

GHOSTING??



Ořezávací frekvence RC filtru:

$$f_c = \frac{1}{2 \cdot \pi \cdot RC}$$

