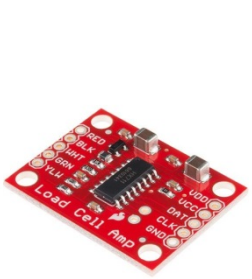
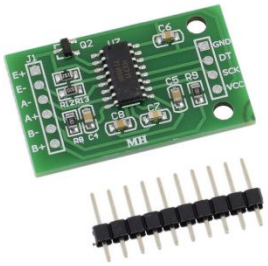
24 bitový dvoukanálový AD převodník s obvodem HX711





Jako vždy, nejdřív si projdeme datasheet obvodu HX711, abychom se dozvěděli, jak obvod přesně pracuje.

Jádrem převodníku je obvod HX711, který obsahuje dva kanály. Vstupní multiplexer volí mezi difernciálními vstupy A a B. Kanál A může být programován se zesílením (GAIN) 128, nebo 64, které koresponduje s rozsahem +-20 mV, nebo +-40 mV, pokud je vstup AVDD připojen k napájecímu pinu. Kanál B má pevně nastavené zesílení 32. Obvod má vlastní stabilizátor napětí. Řídící hodiny obvodu mohou být externí, nebo můžeme využívat interní oscilátor. Kompletní řízení obvodu probíhá prostřednictvím jeho pinů.

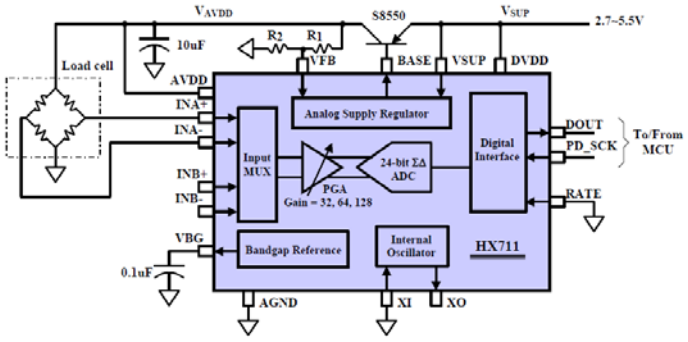
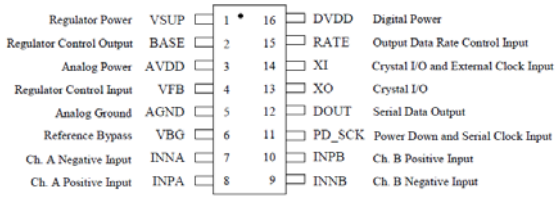


Fig. 1 Typical weigh scale application block diagram



PIN	Jméno pinu	Funkce	Popis
1	VSUP	Napětí	Regulátor napětí 2.5 – 5.5 V
2	BASE	Analogový výstup	Řízení výstupu regulátoru (není-li použito nikam nepřipojit)
3	AVDD	Napětí	Analogové napětí 2.6 – 5.5 V
4	VFB	Analogový vstup	Řízení vstupu regulátoru (není-li použito připojit k AGND)
5	AGND	Zem	Analogová zem
6	VBG	Analogový výstup	Bypass výstupu refernce (1.25V)
7	INA-	Analogový vstup	Kanál A negativní vstup

8	INA+	Analogový vstup	Kanál A pozitivní vstup
9	INB-	Analogový vstup	Kanál B negativní vstup
10	INB+	Analogový vstup	Kanál B pozitivní vstup
11	PD_SCK	Digitalní vstup	Řízení vypnutí napájení (aktivní v H), vstup hodin
12	DOUT	Digitalní výstup	Výstup seriových dat
13	XO	Digitalní I/O	Krystalový oscilátor (není-li použito nikam nepřipojit)
14	XI	Digitalní vstup	Krystalový oscilátor/externí hodiny, 0 – při použití vnitřních hodin
15	RATE	Digitalní vstup	Volba počtu převodů za sec 0: 10 Hz 1:80 Hz
16	DVDD	Napětí	Napájení digitální části obvodu 2.6 – 5.5 V

Plný rozsah napětí diferenciálního vstupu V(inp)- V(inn): +-0.5 (AVDD/GAIN). Kodování výstupních dat (dvojkový doplňkový kód): 800000 – 7FFFFF. Společný režim vstupů je minimálně AGND+1.2V a maximálně AVDD-1.3 V

Vstupy

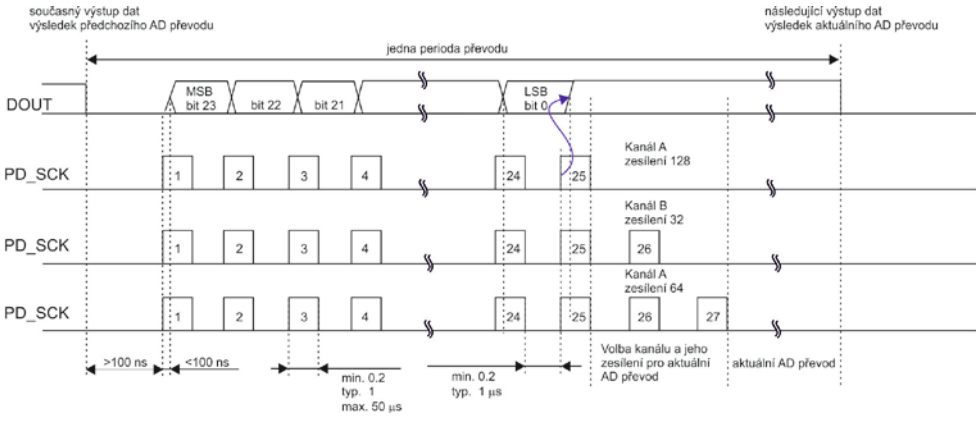
Kanál A je navržen pro měření difernciálního napětí na můstkových senzorech. Může být nastaven se zesílením 128, nebo 64. Velké zesílení je potřebné pro malé výstupní signály můstkových senzorů. Je-li použito napájecí napětí AVDD = 5V, je rozsah měření diferenciálního napětí v závislosti na nastaveném zesílení +-20 mV, nebo +-40 mV.

Kanál B má pevně nastavené zesílení na 32 a rozsah měřených diferenciálních napětí je tedy +-80 mV při napájení AVDD = 5V.

Digitální napájecí napětí DVDD by mělo být stejně velké jako napájení mikrokontroléru. Při použití interního regulátoru napětí jak je nakresleno na obrázku výše je $V_{AVDD} = V_{BG}*(R1 + R2)/R1$ a mělo by být nastaveno na hodnotu minimálně o 100 mV nižší, než je V_{SUP}.

Při použití interního oscilátoru jsou výstupní data vysílána každých 100 ms (10 SPS – převodů za sekundu) je-li RATE = 0, nebo každých 12.5 ms (80 SPS) je-li RATE = 1

Nejsou-li k dispozici naměřená data, je výstup DOUT v H. Vstup hodin PD_SCK by mělo být v L.



Přejde-li výstup DOUT do L, jsou připravena naměřená data k odebrání. Data z převodníku jsou čtena ze výstupu DOUT a jejich přenos je řízen pomocí pozitivních hodin na vstupu PD_SCK. Každý pulz vyzvede jedn bit dat. Data jsou z modulu vysílána od MSB (Nejvýznamnějšího bitu – bit s nejvyšší

váhou) po LSB (Nejméně významný bit – bit s nejnižší váhou), dokud není odebráno všech 24 bitů dat. 25. hodinový puls opět převede DOUT zpět do H.

PD_SCK Pulses	Input channel	Gain
25	A	128
26	B	32
27	A	64

Table 3 Input Channel and Gain Selection

Vstup a zesílení je vybíráno počtem hodinových pulzů, který nesmí být menší než 25 ani větší než 27 v jedné periodě převodu. Jinak dojde k chybě seriové komunikace.

Pokud připojíme obvod k napájecímu napětí, proběhne reset čipu. Pin PD_SCK je použit pro vypnutí obvodu HX711. Je-li PD_SCK v L, obvod normálně pracuje.

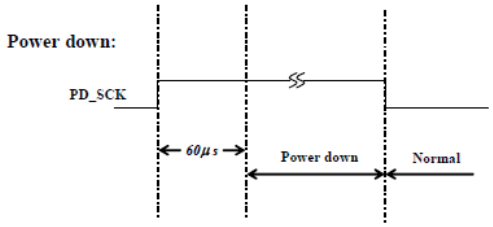


Fig.3 Power down control

Přejde-li PD_SCK do H na dobu delší než 60 μs, přejde obvod do vypnutého stavu (stend byte). Po resetu, nebo opětném přechodu do normálního režimu práce nastaví se automaticky výběr kanálu A se zesílením 128.

Katalogový list obvodu nalezneme třeba zde:
https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf

Převodník je velmi přesný a díky integrovanému zesilovači pracuje i s velmi malými signály (základní rozsahy převodníku +-20 mV a +-40 mV). Jak již bylo řečeno, je velmi vhodný pro měření a převod diferenciálního napětí v můstkových senzorech, jako jsou například tenzometry.

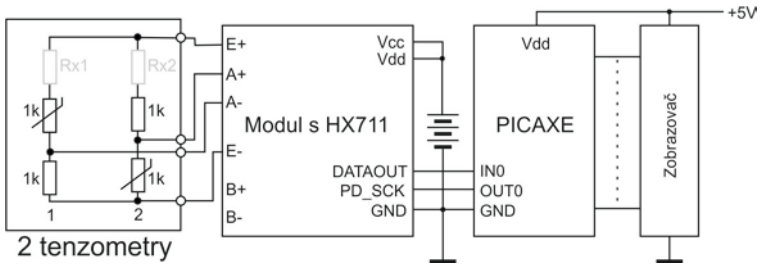
Poznámka: Tenzometry používané v průmyslových aplikacích pro přesná měření jsou poměrně drahé. Nejlevnější se pohybují od 400 Kč výš.

Pro méně náročné aplikace lze použít je senzor ohýbání pro Arduino, model B 120-3AA.

Jeho základní parametry jsou následující:
Rozměry: 6,6 x 3,2 mm, rozměry drátěné mřížky: 3,0 x 2,3 mm, materiál: phenolic-epoxy-acetal, citlivostní koeficient: 2,0 mV/V ± 1%, únavová životnost: > 10 milionů krát, jmenovitá hodnota

odporu: < 3 Ohm, průměrná odchylka odporu: < 0,3 Ohm, koeficient tepelného výkonu: < 2 µm/m/°Cn, mechanické zpoždění: 1,2 um/m.

Činnost modulu můžeme ověřit na jednoduchém příkladu. Jeho zapojení je na následujícím obrázku.



R_{x1} a R_{x2} jsou odpory, kterými dokalibrujeme tenzomatry tak, aby Ud v klidu bylo co nemenší (méně než 0.5 mV).

POZOR! Uvědomme si, že A/D převodník převádí velmi malá napětí 20 – 80 mV a při 24 bitovém převodu, kdy 24. bit je znaménkový, je jeho rozlišení (20 až 80)/2^{22} (4 194 304). To znamená, že rozlišení je +-4.8 nV až +- 19 nV. Jde o extrémně malé signály. Obvod se ve skutečnosti chová jako digitální galvanometr. To vyžaduje dokonalé vyhlazení napájecího napětí. Samostané napájení z baterie tento požadavek splňuje. Připojení vstupních obvodů převodníku musí být co nejblíže k měřícímu obvodu (můstku). Navíc konstrukce celého zařízení (obvody měřícího můstku spolu s přívody) musí být dokonale stíněné proti působení vnějšího rušení a realizované podle zásad pro práci s takto malými signály. V opačném případě nebude možné využít maximální přesnost A/D převodníku.

Zde je příklad podprogramů, pro práci s modulem pomocí procesoru PICAXE 20M2. Procesor má nastavený generátor hodin na 32 MHz.

```
;*****************************************************
; Podprogramy
; CLKO je na C.0, DatIn je na C.1
;*****************************************************
;*****************************************************
; zastaveni cinnosti AD prevodniku - jeho vypnuti
;.....................................................
pwrDownAD:
        high    CLKO
        return
;*****************************************************
; obnoveni cinnosti AD prevodniku - jeho zapnuti
;.....................................................
pwrUpAD:
        low     CLKO
        return
;*****************************************************
; precteni dat z prevodniku
;.....................................................
rdSerData:
        byteDat0=0
        byteDat1=0
        byteDat2=0
;.....................................
        pulsout   CLKO,4                ;precti bit 23
        if DatIn = 0 then goto rds1     ;a uloz ho do bitu 0 datoveho registru
        byteDat2 = byteDat2 OR 1
rds1:
        byteDat2 = byteDat2 + byteDat2  ;posun obsah bytu dat o bit vlevo
        pulsout   CLKO,4                ;precti bit 22
        if DatIn = 0 then goto rds2     ;a uloz ho do bitu 0 datoveho registru
        byteDat2 = byteDat2 OR 1
```

```
rds2:
        byteDat2 = byteDat2 + byteDat2      ;posun obsah bytu dat o bit vlevo
        pulsout   CLKO,4                    ;precti bit 21
        if DatIn = 0 then goto rds3         ;a uloz ho do bitu 0 datoveho registru
        byteDat2 = byteDat2 OR 1
rds3:
        byteDat2 = byteDat2 + byteDat2      ;posun obsah bytu dat o bit vlevo
        pulsout   CLKO,4                    ;precti bit 20
        if DatIn = 0 then goto rds4         ;a uloz ho do bitu 0 datoveho registru
        byteDat2 = byteDat2 OR 1
rds4:
        byteDat2 = byteDat2 + byteDat2      ;posun obsah bytu dat o bit vlevo
        pulsout   CLKO,4                    ;precti bit 19
        if DatIn = 0 then goto rds5         ;a uloz ho do bitu 0 datoveho registru
        byteDat2 = byteDat2 OR 1
rds5:
        byteDat2 = byteDat2 + byteDat2      ;posun obsah bytu dat o bit vlevo
        pulsout   CLKO,4                    ;precti bit 18
        if DatIn = 0 then goto rds6         ;a uloz ho do bitu 0 datoveho registru
        byteDat2 = byteDat2 OR 1
rds6:
        byteDat2 = byteDat2 + byteDat2      ;posun obsah bytu dat o bit vlevo
        pulsout   CLKO,4                    ;precti bit 17
        if DatIn = 0 then goto rds7         ;a uloz ho do bitu 0 datoveho registru
        byteDat2 = byteDat2 OR 1
rds7:
        byteDat2 = byteDat2 + byteDat2      ;posun obsah bytu dat o bit vlevo
        pulsout   CLKO,4                    ;precti bit 16
        if DatIn = 0 then goto rds8         ;a uloz ho do bitu 0 datoveho registru
        byteDat2 = byteDat2 OR 1
rds8:   ;.....                              precten nejvyssi byte dat
        pulsout   CLKO,4                    ;precti bit 15
        if DatIn = 0 then goto rds9         ;a uloz ho do bitu 0 datoveho registru
        byteDat1 = byteDat1 OR 1
rds9:
        byteDat1 = byteDat1 + byteDat1      ;posun obsah bytu dat o bit vlevo
        pulsout   CLKO,4                    ;precti bit 14
        if DatIn = 0 then goto rds10        ;a uloz ho do bitu 0 datoveho registru
        byteDat1 = byteDat1 OR 1
rds10:
        byteDat1 = byteDat1 + byteDat1      ;posun obsah bytu dat o bit vlevo
        pulsout   CLKO,4                    ;precti bit 13
        if DatIn = 0 then goto rds11        ;a uloz ho do bitu 0 datoveho registru
        byteDat1 = byteDat1 OR 1
rds11:
        byteDat1 = byteDat1 + byteDat1      ;posun obsah bytu dat o bit vlevo
        pulsout   CLKO,4                    ;precti bit 12
        if DatIn = 0 then goto rds12        ;a uloz ho do bitu 0 datoveho registru
        byteDat1 = byteDat1 OR 1
rds12:
        byteDat1 = byteDat1 + byteDat1      ;posun obsah bytu dat o bit vlevo
        pulsout   CLKO,4                    ;precti bit 11
        if DatIn = 0 then goto rds13        ;a uloz ho do bitu 0 datoveho registru
        byteDat1 = byteDat1 OR 1
rds13:
        byteDat1 = byteDat1 + byteDat1      ;posun obsah bytu dat o bit vlevo
        pulsout   CLKO,4                    ;precti bit 10
        if DatIn = 0 then goto rds14        ;a uloz ho do bitu 0 datoveho registru
        byteDat1 = byteDat1 OR 1
rds14:
        byteDat1 = byteDat1 + byteDat1      ;posun obsah bytu dat o bit vlevo
        pulsout   CLKO,4                    ;precti bit 9
        if DatIn = 0 then goto rds15        ;a uloz ho do bitu 0 datoveho registru
        byteDat1 = byteDat1 OR 1
rds15:
        byteDat1 = byteDat1 + byteDat1      ;posun obsah bytu dat o bit vlevo
        pulsout   CLKO,4                    ;precti bit 8
        if DatIn = 0 then goto rds16        ;a uloz ho do bitu 0 datoveho registru
        byteDat1 = byteDat1 OR 1
rds16: ;.....                               precten prostredni byte dat
        pulsout   CLKO,4                    ;precti bit 23
        if DatIn = 0 then goto rds17        ;a uloz ho do bitu 0 datoveho registru
        byteDat0 = byteDat0 OR 1
rds17:
        byteDat0 = byteDat0 + byteDat0      ;posun obsah bytu dat o bit vlevo
        pulsout   CLKO,4                    ;precti bit 22
```

```
            if DatIn = 0 then goto rds18            ;a uloz ho do bitu 0 datoveho registru
            byteDat0 = byteDat0 OR 1
rds18:
            byteDat0 = byteDat0 + byteDat0          ;posun obsah bytu dat o bit vlevo
            pulsout   CLKO,4                        ;precti bit 21
            if DatIn = 0 then goto rds19            ;a uloz ho do bitu 0 datoveho registru
            byteDat0 = byteDat0 OR 1
rds19:
            byteDat0 = byteDat0 + byteDat0          ;posun obsah bytu dat o bit vlevo
            pulsout   CLKO,4                        ;precti bit 20
            if DatIn = 0 then goto rds20            ;a uloz ho do bitu 0 datoveho registru
            byteDat0 = byteDat0 OR 1
rds20:
            byteDat0 = byteDat0 + byteDat0          ;posun obsah bytu dat o bit vlevo
            pulsout   CLKO,4                        ;precti bit 19
            if DatIn = 0 then goto rds21            ;a uloz ho do bitu 0 datoveho registru
            byteDat0 = byteDat0 OR 1
rds21:
            byteDat0 = byteDat0 + byteDat0          ;posun obsah bytu dat o bit vlevo
            pulsout   CLKO,4                        ;precti bit 18
            if DatIn = 0 then goto rds22            ;a uloz ho do bitu 0 datoveho registru
            byteDat0 = byteDat0 OR 1
rds22:
            byteDat0 = byteDat0 + byteDat0          ;posun obsah bytu dat o bit vlevo
            pulsout   CLKO,4                        ;precti bit 17
            if DatIn = 0 then goto rds23            ;a uloz ho do bitu 0 datoveho registru
            byteDat0 = byteDat0 OR 1
rds23:
            byteDat0 = byteDat0 + byteDat0          ;posun obsah bytu dat o bit vlevo
            pulsout   CLKO,4                        ;precti bit 16
            if DatIn = 0 then goto rds24            ;a uloz ho do bitu 0 datoveho registru
            byteDat0 = byteDat0 OR 1
rds24: ;.....                                       precten nejnizssi byte dat
            return
;*****************************************************************
; cteni prevodu z kanalu A se zesilenim 128
;.....
rdADA128:
            if DatIn=1 then goto rdADA128           ;cekani na dokonceni prevodu
;.....
            call      rdSerData                     ;volba zesílení 124 a kanálu A
            pulsout   CLKO,4                        ;vyslani 25. hodinoveho pusu
            return
;*****************************************************************
; cteni prevodu z kanalu A se zesilenim 64
;.....
rdADA64:
            if DatIn=1 then goto rdADA64            ;cekani na dokonceni prevodu
;.....
            call      rdSerData
            pulsout   CLKO,4                        ;volba zesílení 64 a kanálu A
            pauseus   1                             ;vyslani 25. hodinoveho pusu
            pulsout   CLKO,4                        ;vyslani 26. hodinoveho pusu
            pauseus   1
            pulsout   CLKO,4                        ;vyslani 27. hodinoveho pusu
            return
;*****************************************************************
; cteni prevodu z kanalu B se zesilenim 32
;.....
rdADB32:
            if DatIn=1 then goto rdADB32            ;cekani na dokonceni prevodu
;.....
            call      rdSerData                     ;volba zesílení 32 a kanálu B
            pulsout   CLKO,4                        ;vyslani 25. hodinoveho pusu
            pauseus   1
            pulsout   CLKO,4                        ;vyslani 26. hodinoveho pusu
            return
```