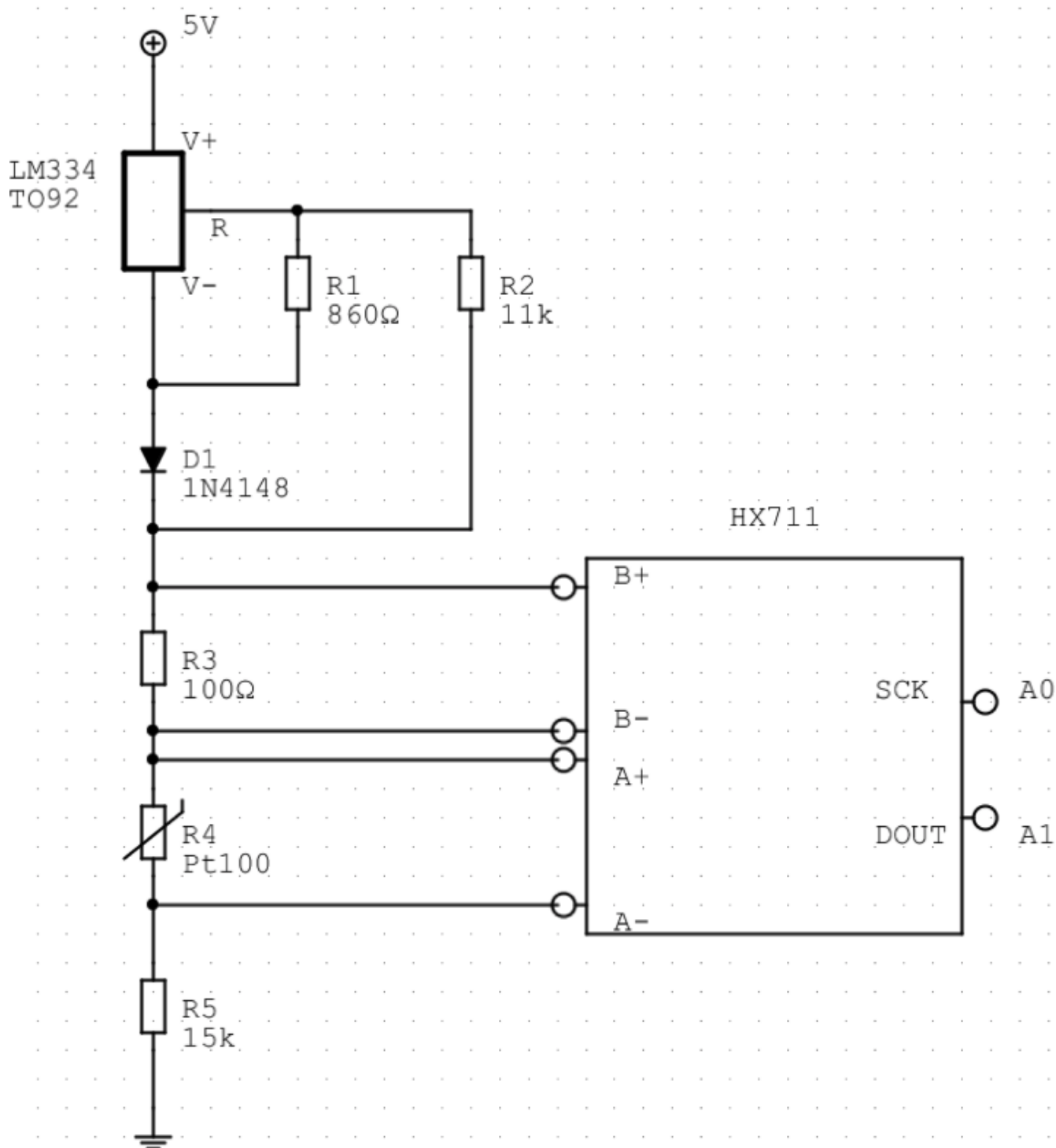


Je zvolena metoda měření úbytku napětí na odporu při napájení konstantním proudem. Zdroj proudu je realizovaný pomocí LM334. Napětí měříme modulem [HX711](#). Pomocí pár součástek získáme jednoduchý, levný (asi 100 Kč) a přesný převodník pro Pt100.



Zdroj konstantního proudu vychází ze zapojení v datasheetu. Tam nalezneme vše potřebné pro případnou úpravu zapojení. Při hodnotách R1,R2 teče proud 168 μ A. Zdroj proudu je napájený 5V.

U modulu HX711 jsou použity vstupy A a B. Napájení E+,E- není použito. Výstup je propojený se vstupy A0 a A1 modulu Arduino. Kanál B slouží pro měření proudu. Můžeme tak sledovat teplotní a časovou závislost zdroje proudu. Případný rozdíl mezi hodnotou uloženou při nastavení převodníku a hodnotou aktuální lze využít pro posun rozsahu. Kanál A používá zesílení 64 a měří úbytek napětí na Pt100. Tato hodnota je v programu převedena na ohmy a následně na teplotu. Na kanálu A při zesílení 64 lze měřit do 370 °C ($238\Omega * 168 \mu A = 0,0399mV$).

K nastavení a testování slouží jednoduchý program. Nastavení pro rozsah 0 - 211°C provedeme pomocí odporu 100 Ω a 180 Ω (přesnost 0,1%). Při 100 Ω zapíšeme konstanty *proud* a *spodni*. Při 180 Ω konstantu *horni*. Hodnota *privod* je trvalý rozdíl mezi měřením Pt100 a senzorem DS18B20(při nastavení *privod* = 0).

Potřebná knihovna HX711 je ke stažení [zde](#), návod pro její importování [zde](#).

```
kopírovat do schránky#include "HX711.h"

#include <OneWire.h>
#include <DallasTemperature.h>
#include <Nokia5110.h>

// nastavení čísla vstupního pinu
const int pinCidlaDS = 2;

// vytvoření instance oneWireDS z knihovny OneWire
OneWire oneWireDS(pinCidlaDS);
// vytvoření instance senzoryDS z knihovny DallasTemperature
DallasTemperature senzoryDS(&oneWireDS);

HX711 scale;

#define Ka 0.0039083
#define Kb -5.775E-07
#define proud 1679.5 //hodnota vst. B pri nastaveni
#define spodni 3346.0 //hodnota vst. A pri 100 Ohmech
#define horni 6016.0 //hodnota vst. A pri 180 Ohmech
#define privod 0.4 //kompenzace privodu
```

```
// nastavení propojovacích pinů
#define RST 10
#define CE 9
#define DC 8
#define DIN 7
#define CLK 6

// inicializace LCD displeje z knihovny
LCDnokia5110 lcd(RST, CE, DC, DIN, CLK);

String zapis = "";
char tisk[10];
float spodni_a;
float horni_a;

void setup() {

    // zahájení komunikace s displejem
    lcd.LcdInitialise();
    // vymazání obsahu displeje
    lcd.LcdClear();
    // nastavení velikosti mezery mezi písmeny,
    // lze nastavit na 0, 1 či 2 body
    lcd.CharSpace = 1;

    lcd.GotoXY(20,0);
    lcd.LcdString("start");

    senzoryDS.begin();
```

```

    // parameter "gain" is omitted; the default value 128 is used by the
    library
    // HX711.DOUT      - pin #A1
    // HX711.PD_SCK    - pin #A0
    scale.begin(A1, A0, 32);
    delay(2000);
    //zahrati systemu
    float z = scale.read_average(20);
    scale.set_gain(64);
    z = scale.read_average(20);

}

```

```

void loop() {

```

```

//***** teplota DS18B20 *****

```

```

    senzoryDS.requestTemperatures();
    float teplota_DS = senzoryDS.getTempCByIndex(0);

```

```

//***** cteni proudu *****

```

```

    scale.set_gain(32);
    float proud_a = scale.read_average(3)/1000.0;

```

```

    //omezeni na 1 des. misto o hodnotu 0 nebo 0.5

```

```

    int prumer = int(proud_a);
    float zbytek = proud_a - float(prumer);
    if (zbytek < 0.5){
        proud_a = float(prumer);
    }

```

```

    }
    else {
        proud_a = float(prumer) + 0.5;
    }

    float podil = proud_a / proud;

    /****** cteni odporu *****/
    scale.set_gain(64);
    //uprava zacatku rozsahu podle zmeny proudu
    spodni_a = spodni * podil;
    //uprava konce rozsahu podle zmeny proudu
    horni_a = horni * podil;

    //omezeni des. míst
    prumer = int(scale.read_average(3)/1000.0);
    float x =float(prumer);

    //prepocet na odpor 100 - 180 ohmu
    float odpor = (x - spodni_a) * (180.0 - 100.0) / (horni_a - spodni_a)
+ 100.0 ;

    //prepocet na teplotu 0 - 211 st.
    float teplota = (( sqrt(Kb * odpor - 100.0 * Kb + 25.0 * pow(Ka,2))
- 5.0 * Ka) / (10.0 * Kb)) - privod;

    /****** zapis hodnot *****/
*
    //mereni proudu vstup B
    lcd.GotoXY(20,0);
    zapis = String(proud_a,DEC);
    zapis.toCharArray(tisk, 8);
    lcd.LcdString(tisk);

```

```
//mereni Pt100 vstup A
lcd.GotoXY(20,1);
zapis = String(x,DEC);
zapis.toCharArray(tisk, 8);
lcd.LcdString(tisk);

//prepocet na odpor
lcd.GotoXY(20,2);
zapis = String(odpor,DEC);
zapis.toCharArray(tisk, 8);
lcd.LcdString(tisk);

//prepocet na teplotu
lcd.GotoXY(20,3);
zapis = String(teplota,DEC);
zapis.toCharArray(tisk, 8);
lcd.LcdString(tisk);

//teplota DS18B20
lcd.GotoXY(20,4);
zapis = String(teplota_DS,DEC);
zapis.toCharArray(tisk, 8);
lcd.LcdString(tisk);

delay(5000);

}
```