

# SIMPLE EMBEDDED STEPPER MOTOR DRIVER FOR INDUSTRIAL USAGE

Lucie Byrtusová

Bachelor Degree Programme (3), FEEC BUT

E-mail: xbyrtu03@stud.feec.vutbr.cz

Supervised by: Ondřej Baštán

E-mail: xbasta02@stud.feec.vutbr.cz

**Abstract:** Paper describes the possibility to control unipolar stepper motor from PLC using cheap microcontroller ATMEGA328P and 16-Bit I/O expander MCP23017. First is described the driver and explained the principals of I2C serial bus. Further is described microchip MCP23017 and it's registers essential for this work. In the end are shown examples how to program the controller via environment Arduino Studio.

**Keywords:** I2C, ATMEGA328P, PLC, Arduino Studio, expander, MCP23017, unipolar motor

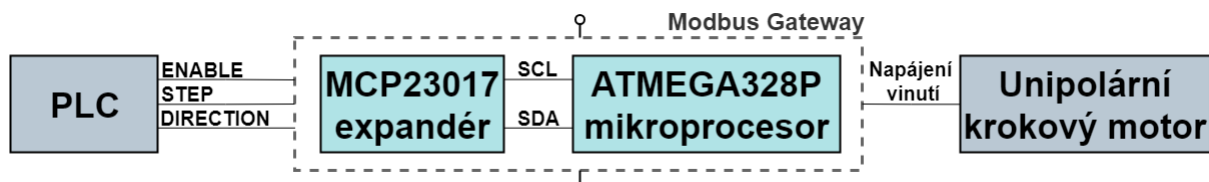
## 1 ÚVOD

Výrobní buňka Sklad skleniček je součástí testbedu zvaného Barman, který byl navržen pro demonstraci výrobního procesu odpovídajícího předpokladům Průmyslu 4.0. Úkolem Skladu skleniček je jednak mít k dispozici zásobu čistých skleniček a jednak ukládat špinavé skleničky, včetně sběru dat o ukončeném výrobním procesu.

Pro manipulaci se skleničkou má buňka k dispozici manipulační aparát (viz. Obrázek 4(b)), jehož pohyb v horizontální rovině ovládá unipolární motor, který pracuje s nižší logickou úrovní (5V), než je typická pro průmyslová zařízení. Motor je řízen signály z PLC pomocí jednoduchého externího budiče (viz. Obrázek 4(a)), který s PLC komunikuje prostřednictvím standartních řídicích signálů, a pro řízení svých výstupů využívá sběrnici I2C. Obsahem tohoto příspěvku je popis realizace zmíněného řízení.

## 2 POPIS KONFIGURACE

**Modbus Gateway** je hardware vytvořen za účelem komunikace průmyslových zařízení a zastupuje zde roli driveru. Vzhledem k nedostatku volných vstupně-výstupních pinů MCU je pro rozšíření implementován 16bitový expandér **MCP23017**. Expandér komunikuje po I2C sběrnici s mikroprocesorem **ATMEGA328P**, ve kterém se modifikují data tak, aby byla správně předána motoru. Z PLC přicházejí tři signály udávající povolení chodu motoru, směr chodu a počet kroků. Motoru jsou tyto signály, spínané pomocí tranzistorového pole **ULN2003**, interpretovány jako postupné napájení čtyř jednotlivých vinutí motoru.



Obrázek 1 Blokové schéma konfigurace

### 3 SBĚRNICE I2C

I2C (Internal-Integrated-Circuit Bus) je dvojdátová datová sběrnice využívaná pro sériovou komunikaci a přenos dat. Vhodné použití nachází tam, kde nevznikají velké nároky na rychlost, ale je potřeba snížit počet vodičů.

Prvním z vodičů je SDA (Serial Data) linka, jejímž účelem je přenos dat. Druhým vodičem je SCL (Serial Clock). Rychlost přenosu závisí na zvolené frekvenci hodin, přičemž základní frekvence je 100 kHz.

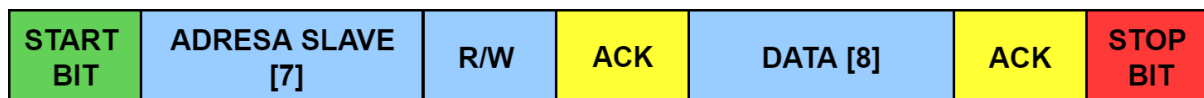
Komunikace je typu Master-Slave, kde master je ve většině případů používán mikrokontrolér. Obvody umožňují zapojení jak více slávů, tak také více masterů. Při vysílání přijímají data všichni účastníci. Až na základě adresy určují, jestli jsou data určena pro ně a budou s nimi operovat.

#### 3.1 PRŮBĚH PŘENOSU

Master iniciuje přenos vysláním **START** bitu, tedy změnou úrovně SDA z 1 na 0. Následuje příjem 1 Bytu dat upřesňující komunikaci. 7 bitů reprezentuje adresu oslovovaného zařízení. Poslední, 8. bit **READ/WRITE** definuje, zda se bude jednat o zápis či čtení.

Po konfiguraci zahajuje slave, skrze první acknowledge bit, vysílání a může dojít k samotnému přenosu dat po 1 Bytu. Maximální počet přenesených Bytů není stanoven.

**ACKNOWLEDGE** bit generuje přijímací zařízení. Bit je vyslán po každém přijatém Bytu informace. Pokud byl přenos úspěšný, odešle přijímač logickou 0, což je signál, že je přijímač připravený přijmout další byte. V případě selhání přenosu odesílá přijímač logickou 1 a master ukončuje přenos generováním **STOP** bitu.



Obrázek 2 Průběh komunikace po sběrnici I2C

### 4 EXPANDÉR MCP23017

Expandér má pro účely sběrnice čtyři z adresních bitů zadány neměnně. Adresa je zadávána pomocí zbylých tří bitů, což umožňuje připojení celkem osmi různých zařízení na sběrnici.

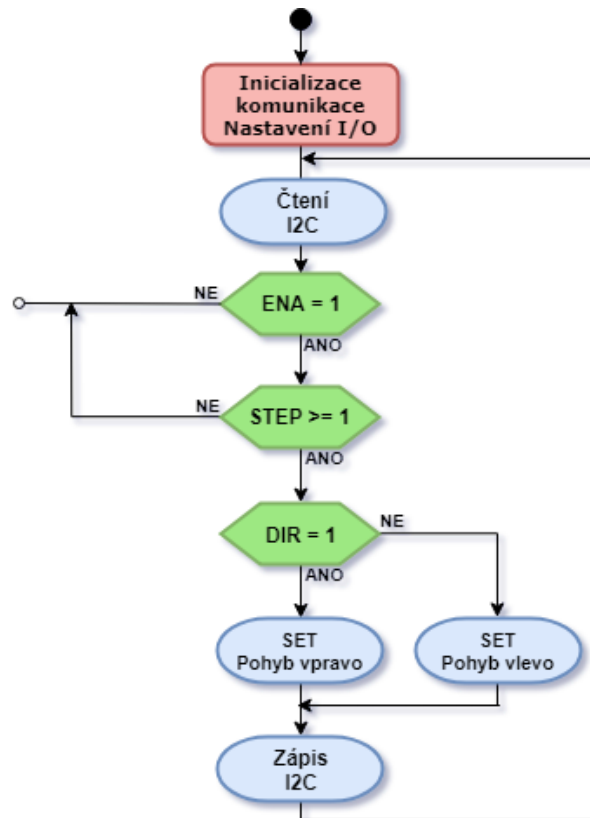
Porty mikročipu jsou rozděleny na porty A a porty B, jejichž účel může být modifikován v **I/O DIRECTION** registru. Logická 1 nastavuje port jako vstupní, logická 0 jako výstupní.

Samotnou hodnotu portu reflektuje **GPIO PORT** registr. Čtení z tohoto registru znamená čtení reálných logických hodnot na I/O pinu. Ovšem pro zápis je vhodnější použít registr **Output Latch**, neboť klopné obvody mohou způsobovat, že když budeme, např. při náhodném zkratování pinu k zemi, zapisovat do OLAT registru log. 1, tak zpětně bude číst OLAT původní nastavenou hodnotu, ale GPIO PORT bude číst log. 0, neboť pin je stále uzemněn.

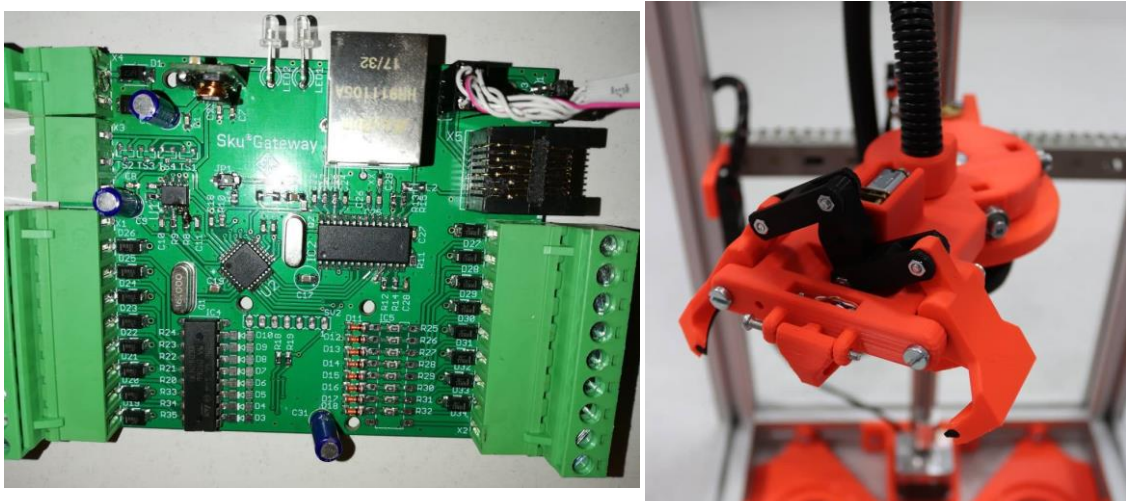
### 5 IMPLEMENTACE V ARDUINO STUDIO

Na obrázku 2 je znázorněn vývojový diagram, dle kterého byl software implementován. Prvním krokem je nastavení portů A, na které jsou přivedeny vodiče motorů, jako výstupní. Porty B se nastaví jako vstupní, neboť jsou na ně přivedeny řídicí signály z PLC. Sekce pro inicializaci je v Arduino studio označována jako *setup* a zavolá se jednou při prvním spuštění. V této sekci je rovněž vytvořen byte M, který se bude odesílat jako data byte přes sběrnici a změnou těchto osmi bitů bude ovládn motor.

Druhou sekci je **loop** a volá se cyklicky po doběhnutí překladače na konec oddílu. V sekci se nejprve vyčte data byte z portu B, který bude obsahovat informace signálů ENABLE, STEP a DIRECTION z PLC. Na základě těchto informací se bude modifikovat byte M, dle splnění podmínek. V případě, že není signálem ENABLE povolen chod motoru, vyskočí program z cyklu a zahájí nové čtení. Stejně tak se stane, pokud je nulový signál STEP, nastavující počet kroků motoru. Třetí podmínka je pro stanovení směru, kterým motor provede daný počet kroků. Takto modifikovaný data byte je připraven pro zápis na výstupní porty a po provedení zápisu běží nový cyklus.



Obrázek 3 Vývojový diagram programování procesoru ATMEGA328P



Obrázek 4 (a) Modbus Gateway a (b) detail manipulátoru

## 5.1 PŘÍKLAD KÓDU A JEHO ROZBOR

Byte, jak datový, tak konfigurační se pro zpřehlednění kódu zapisuje v hexadecimální soustavě. Pro práci s expandérem a sběrnicí I2C je k dispozici knihovna Wire.h.

Na obrázku 4(a) je nejprve pomocí příkazu *begin()* povolena komunikace přes I2C. Příkazem *beginTransmission(0x20)* se zahajuje komunikace se zařízením, jehož adresa 0x20 je zadána binární hodnotou třech adresních pinů 000. Dalším krokem je výběr registru, se kterým budeme pracovat. Je nutné nastavit porty A jako výstupní. Z datasheetu se vyčte adresa registru, příkazem *write(0x00)* se do něj nastaví pointer registrů a stejným příkazem se nastaví jako výstup. Příkazem *endTransmission()* se ukončuje operace.

Na obrázku 4(b) je příklad čtení z registru. Nejprve již známým příkazem posuneme pointer na správný registr, v tomto případě GPIO, ze kterého chceme vyčíst hodnoty na vstupních pinech. Pro čtení se používá příkaz *requestFrom(0x20, 2)*, ve kterém adresujeme komunikované zařízení a počet bytů, které chceme vyčíst. Vyčtené byty je nutné uložit do proměnných typu byte příkazem *read()*.

```
#include <Wire.h>

void setup() {
  Wire.begin();

  Wire.beginTransmission(0x20); //device adress
  Wire.write(0x00); //IODIRA register
  Wire.write(0x00); //output set
  Wire.endTransmission();
}

void loop() {
  Wire.beginTransmission(0x20);
  Wire.write(0x13); //GPIOB register
  Wire.endTransmission();

  Wire.requestFrom(0x20, 2); //reading 2 bytes
  tmp1 = Wire.read(); //storing first byte
  tmp2 = Wire.read(); //storing second byte
}
```

Obrázek 5 (a) Příklad zápisu do registru (b) Příklad čtení z registru

## 6 ZÁVĚR

V příspěvku je popsáno řešení problému s nedostupností driverů neprůmyslových zařízení na trhu. V případě buňky Sklad skleniček je neprůmyslovým zařízením unipolární motor ovládající manipulátor. Sestava určena k jeho řízení je složena z mikroprocesoru ATMEGA328P, expandéru MCP23017 a tranzistorového pole ULN2003 a je součástí hardwaru pro komunikaci různých zařízení, zvaného Modbus Gateway.

Motor je řízen třemi signály z PLC, ENA, STEP, DIR, které jsou skrze I2C sběrnicí překládány na datové byty a poté posílány do příslušných registrů expandéru. Čtení a zápis těchto registrů je implementován v prostředí Arduino Studio užitím knihovny Wire.h.

Při správné implementaci již nemusí uživatel dále zasahovat do programu mikrokontroléru a může ovládat motor z libovolného PLC třemi signály. Alternativním řešením by mohlo být propojení mikrokontroléru skrze protokol Modbus, jehož náročnost aplikace se však liší dle zvoleného PLC a řešení není přenositelné.

## REFERENCE

- [1] MICROCHIP TECHNOLOGY INC. MCP23017/MCP23S17 Datasheet: 16-Bit I/O Expander with Serial Interface. 2007.
- [2] Stručný popis sběrnice I2C a její praktické využití. Vyvoj.hw.cz [online]. [cit. 2019-03-12]. Dostupné z: <https://vyvoj.hw.cz/navrh-obvodu/strucny-popis-sberrnice-i2c-a-jeji-prakticke-vyuziti-k-pripojzeni-externi-EEPROM-24lc256>
- [3] BOXALL, John. Tutorial: Arduino and the I2C bus – Part One. Tronixstuff.com [online]. [cit. 2019-03-12]. Dostupné z: <https://tronixstuff.com/2010/10/20/tutorial-arduino-and-the-i2c-bus/>