



Reference

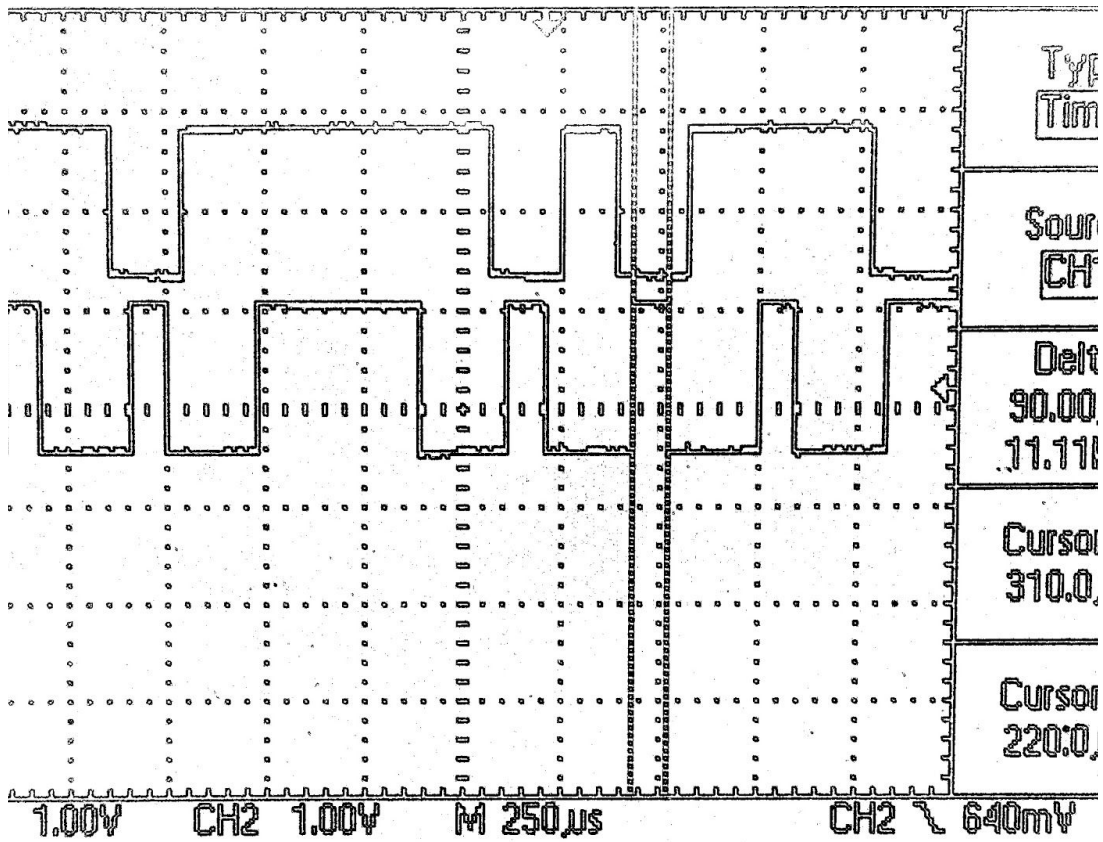
- [1] K20 Sub-Family Reference Manual, Doc. No.: K20P100M100SF2V2RM, Freescale, Jun 2012.
- [2] Universal Serial Bus Specification, Revision 2.0, April 2000.
- [3] NIST/SEMATECH e-Handbook of Statistical Methods, <http://www.itl.nist.gov/div898/handbook/>, aktualizace 30.10.2013
- [4] TAYLOR HOBSON PRECISION, A guide to the Measurement of Roundness, http://www.tarkkuustuonti.fi/Kampanjat/Brochure_Roundness_Booklet.pdf, verze 26.5.2016
- [5] (kód) ControlsFX 8.4.10, <http://controlsfx.bitbucket.org/org/controlsfx/control/CheckComboBox.html>
- [6] (nástroj) WaveDrom <http://wavedrom.com/editor.html>, verze 26.5.2016
- [7] (nástroj) JGraph Ltd. Draw, <http://draw.io>, verze 26.5.2016

Přílohy

1	Osciloskop - hrana hodin	32
2	Osciloskop - formát dat.....	33
3	Osciloskop - délka datového řetězce	34
4	Osciloskop - sgnum	34
5	Inicializace SPI modulu	35
6	Procedura modulu PDB	36
7	Procedura úvodního menu	37
8	Vytvoření datové struktury	37
9	Vytvoření listu GraphPoint objektů	38
10	Schéma základní desky	39
11	Schéma zapojení procesoru	40
12	Fotka výsledného zařízení	41

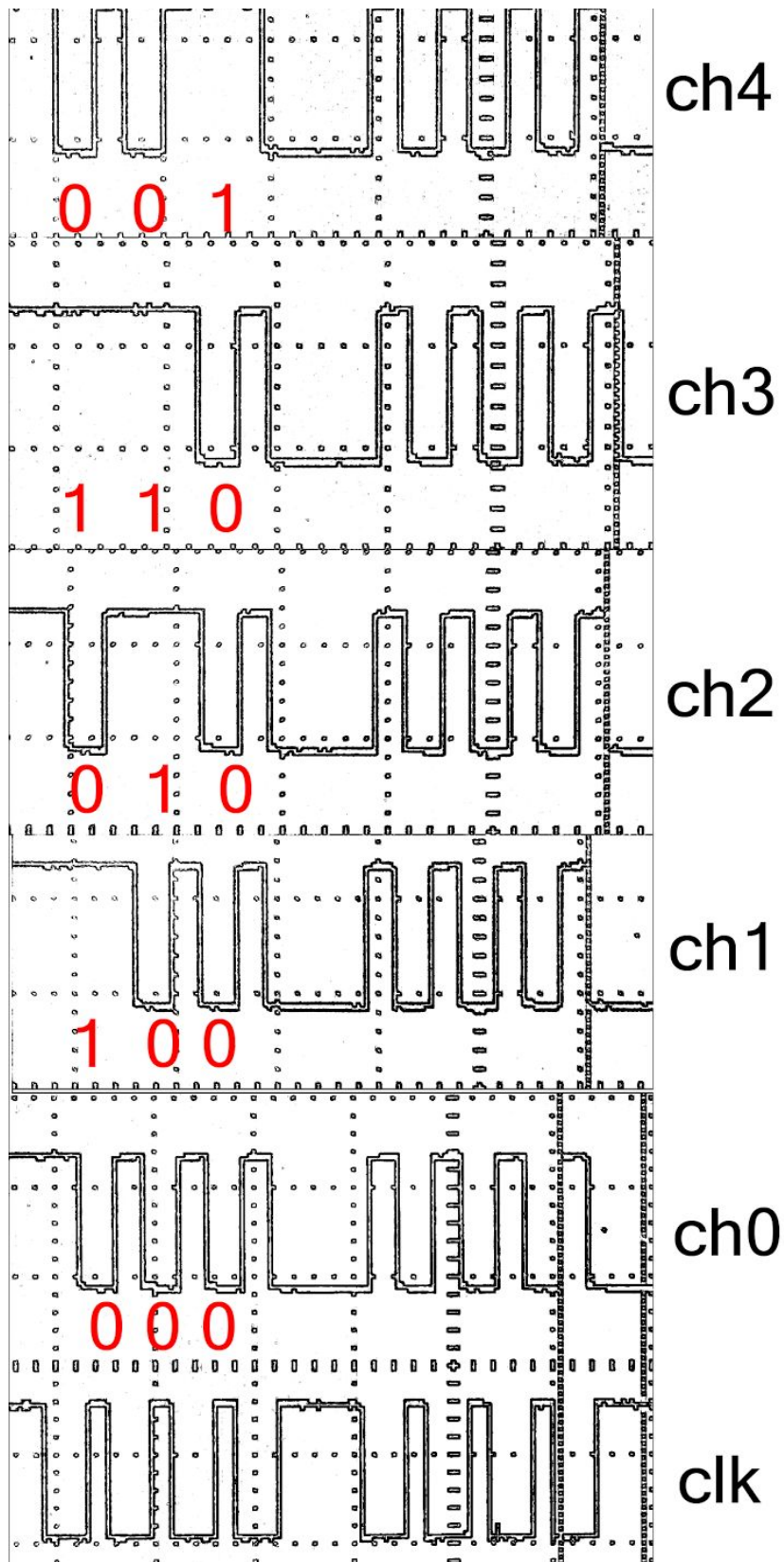


Příloha 1 - Osciloskop - hrana hodin



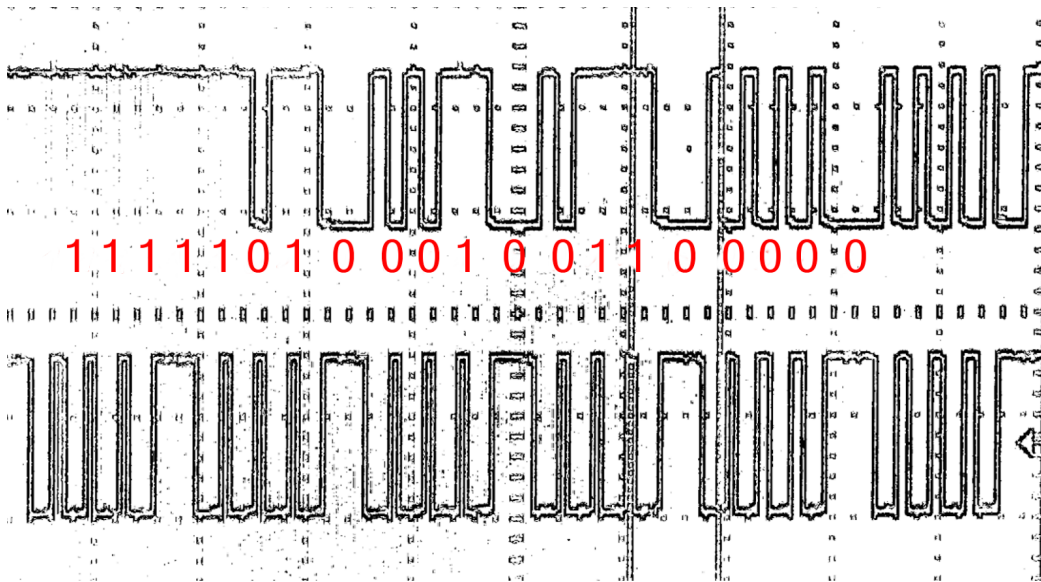


Příloha 2 - Osciloskop - formát dat

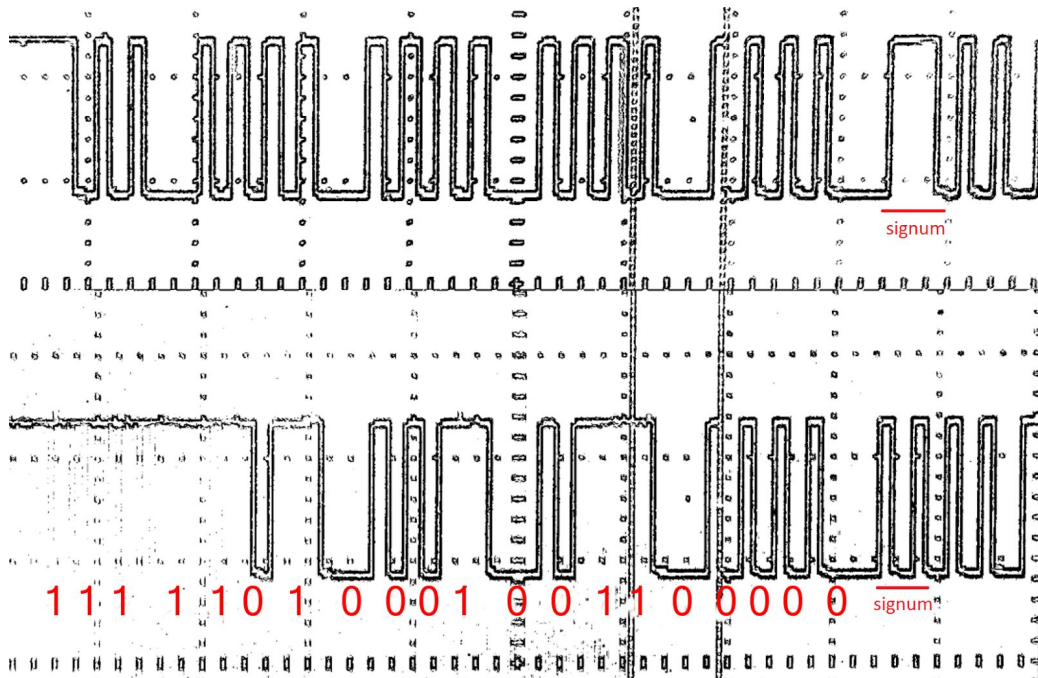




Příloha 3 - Osciloskop - délka datového řetězce



Příloha 4 - Osciloskop - signum



**Příloha 5 - Inicializace SPI modulu**

```
1. // Zastaví modul SPI
2. SPI0_MCR = SPI_MCR_HALT_MASK;
3.
4. // Vyprázdní FIFO
5. SPI0_MCR = (SPI_MCR_CLR_TXF_MASK | SPI_MCR_CLR_RXF_MASK |
   SPI_MCR_HALT_MASK );
6.
7. // Zastaví modul během ladění | Nastavení hodin
8. SPI0_MCR = ( SPI_MCR_FRZ_MASK | SPI_MCR_CONT_SCKE_MASK);
9.
10. // Nuluje počítadlo provedených přenosů
11. SPI0_TCR = 0x00U;
12.
13. // Nastaví sondu jako master zařízení
14. SPI0_CTAR0 = SPI_CTAR_SLAVE_FMSZ(21);
15.
16. // Hodiny jsou mimo přenos v 1
17. SPI0_CTAR0 |= SPI_CTAR_SLAVE_CPOL_MASK;
18.
19. // Vynulování stavového registru
20. SPI0_SR = SPI_SR_TCF_MASK |
21.     SPI_SR_EOQF_MASK |
22.     SPI_SR_TFUF_MASK |
23.     SPI_SR_TFFF_MASK |
24.     SPI_SR_RFOF_MASK |
25.     SPI_SR_RFDF_MASK;
26.
27. // Zakáže SPI modulu vyvolat systémová přerušení
28. SPI0_RSER = 0;
29.
30. // Nastavení pinů
31. PORTC_PCR4 = (uint32_t)(0x00000204UL); // CS
32. PORTC_PCR5 = (uint32_t)(0x00000204UL); // CLK
33. PORTC_PCR7 = (uint32_t)(0x00000204UL); // IN
```

**Příloha 6 - Procedura modulu PDB**

```
1. GPOUT_OFF_SPICS; // Nastaví chip-select na 0
2.
3. // Bit vyvolávající přerušeni pro tuto proceduru
4. // shazuji zpět do 0
5. PDB0_SC &= ~PDB_SC_PDBIF_MASK;
6.
7. // Čtu stav čítače (délku poslední prodlevy)
8. // Pokud je menší než nula, jedná se o začátek datového přenosu
9. if (PDB0_IDLY < 0x150) {
10.  PDB0_IDLY = 0xc000; // nastavuji délku příští prodlevy
11.
12.  // Čtení dat z FIFO
13.  SPI_KM_d1 = SPI0_POPR & 0x3F;
14.  SPI_KM_d2 = SPI0_POPR & 0x3F;
15.  SPI_KM_d4 = SPI0_POPR & 0x3F;
16.
17.  // Poskládání celého datového segmentu
18.  SPI_KM_d1 = SPI_KM_d1>>26;
19.  SPI_KM_d1 |= SPI_KM_d2>>20;
20.  SPI_KM_d1 |= SPI_KM_d3>>14;
21.  SPI_KM_d1 |= SPI_KM_d4>> 8;
22.
23.  //Data uložíím do proměnné
24.  Deviation = SPI_KM_d1;
25.  // Zastavení modulu SPI
26.  SPI0_MCR = SPI_MCR_HALT_MASK;
27.  // Nulování FIFO modulu SPI
28.  SPI0_MCR = (SPI_MCR_CLR_TXF_MASK | SPI_MCR_CLR_RXF_MASK |
29.              SPI_MCR_HALT_MASK);
30.  SPI0_MCR = ( SPI_MCR_FRZ_MASK);
31. }
32. // Konec datového přenosu
33. else {
34.  GPOUT_ON_SPICS; // Chip-select do 1
35.  PDB0_IDLY = 0x100; // Nastaví krátkou příští prodlevu
36. }
37. // Zápis parametrů PDB modulu
38. PDB0_SC |= PDB_SC_LDOK_MASK;
```




Příloha 7 - Procedura úvodního menu

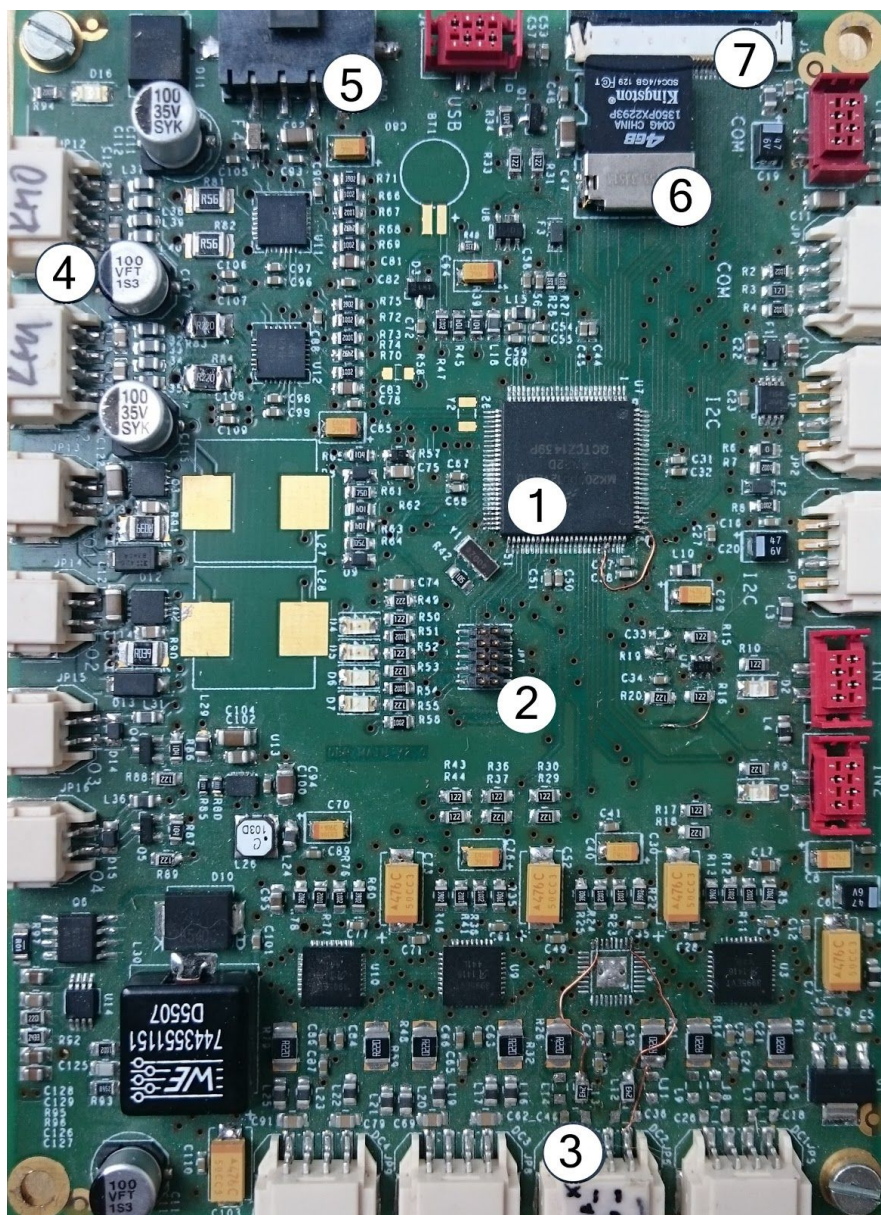
```
1. /**
2.  * Procedura obsluhující úvodní menu
3.  */
4. void startMenu(void) {
5.     // Pokud došlo ke stisku, uloží do key id tlačítka
6.     key = Kom_Zona_Uk->KZ_TSI_TestKey(TSI_FIRE_ON);
7.
8.     printDeviation(); // Vytiskne na display současnou odchylku polohového senzoru
9.
10.    if (!(GPIOE_PDIR & 0x2000000)) { // Pokud je magnetický senzor aktivní
11.        Kom_Zona_Uk->KZ_AjsPan_draw(255,112,"a:\\BMP\\magnet_on.bmp");
12.    }
13.    else { // Pokud aktivní není
14.        Kom_Zona_Uk->KZ_AjsPan_draw(255,112,"a:\\BMP\\magnet_off.bmp");
15.    }
16.
17.    switch (key) {
18.        case 0x01: // Tlačítko start
19.            Kom_Zona_Uk->KZ_AjsPan_draw(0,0,"a:\\BMP\\measuring.bmp");
20.            Mloop_Proc = sessionMenu;
21.            break;
22.        case 0x02: // Tlačítko USB
23.            Kom_Zona_Uk->KZ_AjsPan_draw(0,0,"a:\\BMP\\usb.bmp");
24.            Mloop_Proc = usbMenu;
25.            break;
26.    }
27. }
```

Příloha 8 - Vytvoření datové struktury

```
1. File[] listOfFiles = folder.listFiles();
2.
3. for(int i=0; i<listOfFiles.length; i++){
4.     if( listOfFiles[i].isFile()
5.         && listOfFiles[i].getName().toLowerCase().startsWith("data")){
6.         if(session == null){
7.             session = new Session();
8.         }
9.         session.addMeasurement(listOfFiles[i].getAbsolutePath());
10.    }
11. }
```

**Příloha 9 - Vytvoření list `GraphPoint` objektů**

```
1. // Create graph points
2.     if (graphPoints.size() == id) {
3.         graphPoints.add(new ArrayList<GraphPoint>());
4.         for (int i = 0; i < measurement.LENGTH; i++) {
5.             int x = (int) (i * xScale + Dimensions.BORDER_GAP);
6.             int y = (int) ((session.maxScore - measurement.data.get(i)) *
7.                 yScale / 2 + Dimensions.BORDER_GAP);
8.
9.             GraphPoint gp;
10.            if (Double.compare(measurement.max(), measurement.data.get(i)) == 0
11.                || Double.compare(measurement.min(), measurement.data.get(i)) == 0)
12.            {
13.                gp = new GraphPoint(x, y, false, true, measurement.data.get(i));
14.                extremes++;
15.            } else {
16.                gp = new GraphPoint(x, y, false, false, measurement.data.get(i));
17.            }
18.
19.            graphPoints.get(id).add(gp);
20.        }
21.        linePoints = graphPoints.get(id);
22.    } else {
23.        linePoints = graphPoints.get(id);
    }
```


Příloha 10 - Schéma základní desky

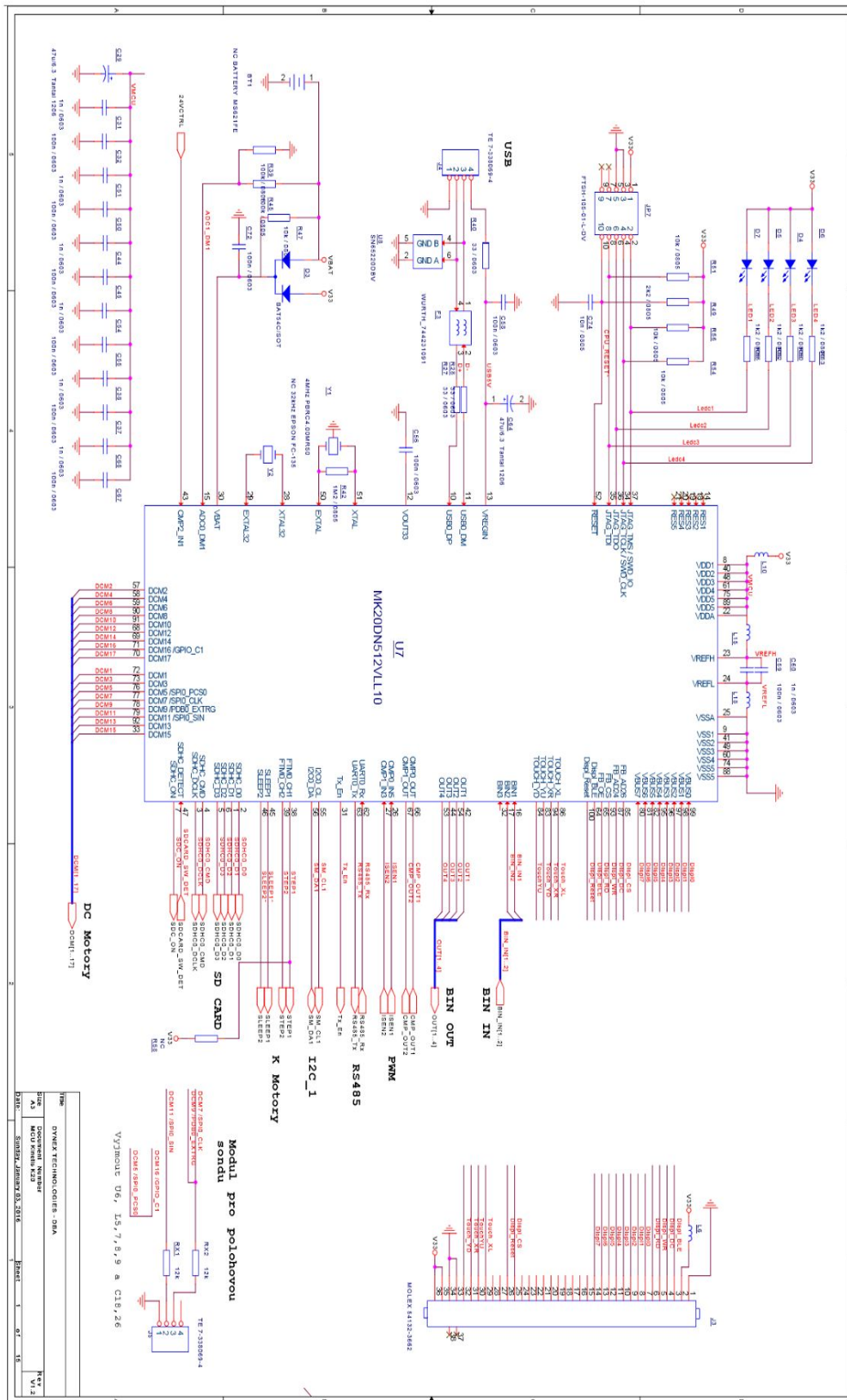
Obrázek p.1 Popis základní desky

Popis pro tento projekt důležitých součástí základní desky:

1. ARM procesor
2. Konektor programovacího rozhraní JTAG
3. Upravený vstup pro polohový senzor
4. Výstupy pro krokové motory
5. Napájecí konektor
6. Čtečka microSD karet
7. Konektor pro připojení dotykové obrazovky



Příloha 11 - Schéma zapojení procesoru



Příloha 12 - Fotka výsledného zařízení

