AN039: TMC5072-EVAL and Arduino via Single Wire UART

Document Revision V0.10 • 2017-Mar-03

This application note describes how to connect TRINAMIC's TMC5072-EVAL via Single Wire UART to an Arduino Mega for basic operation. The wiring is limited to the basic functionality to communicate via Single Wire UART.

Contents

1 Preparation

To use the 5V version of the Arduino MEGA you have to resolder Resistor from position R3 to R8. This enables 5V logic level for the TMC5072. The sense resistor by default is soldered to the right position. Please desolder and resolder to the left position.

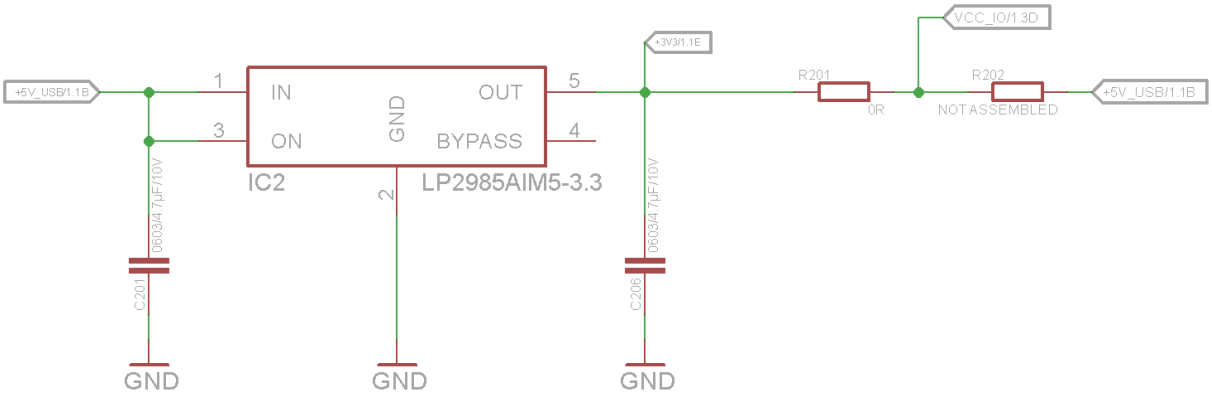

Figure 1: Extract from TMC5072-EVAL schematic

©2017 TRINAMIC Motion Control GmbH & Co. KG, Hamburg, Germany
Download newest version at: www.trinamic.com

TRINAMIC
MOTION CONTROL

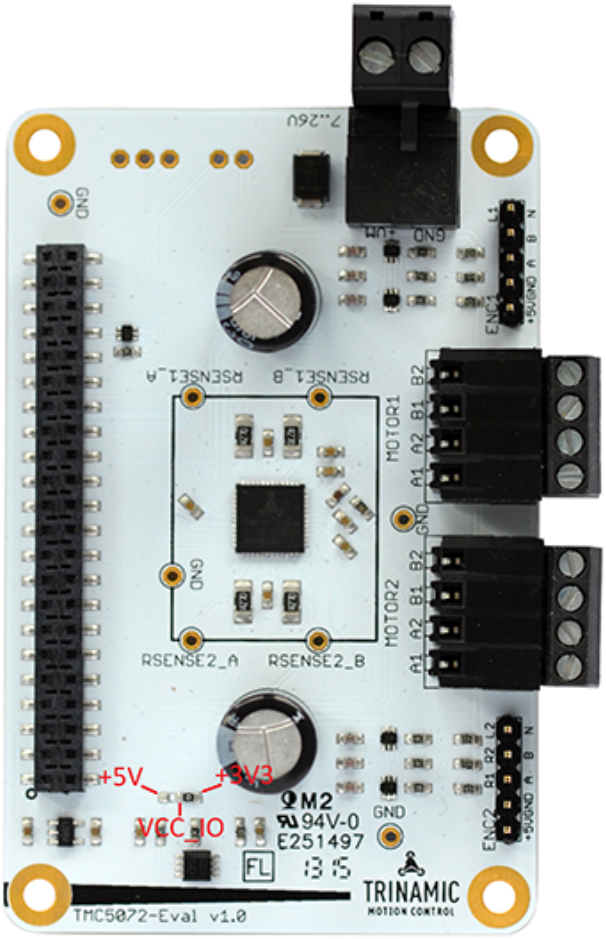

Figure 2: TMC5072-EVAL with pins marked for logic supply voltage

2 Wiring

The wiring is very simple. You will need X jumper wires. To make the wiring easier you can connect the wire directly to the Eselsbrücke. As a reference you can use the TMC5072Eval_v10_01_Schematic.pdf. Here you find the signals that are on each pin. The configuration is documented in the comment section of the Arduino code.

©2017 TRINAMIC Motion Control GmbH & Co. KG, Hamburg, Germany
Terms of delivery and rights to technical change reserved.
Download newest version at www.trinamic.com
Read entire documentation.

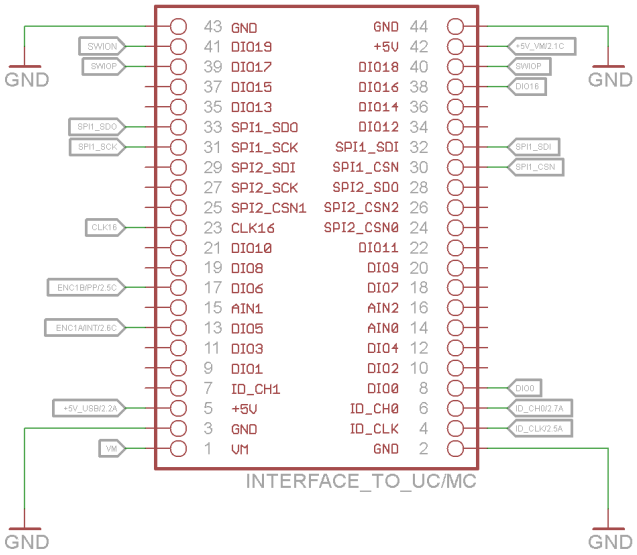

Figure 3: Pinheader of TMC5072-EVAL

3 Arduino Code

The Arduino Code below is based on the TMC-API package. The program initializes the TMC5072 and executes a simple move to position cycle. It will rotate two 200 full step motors 10 revolutions to the one and 10 revolutions to the other direction depending on the wiring of the stepper motor. Please refer to the TMC5072 datasheet or the TMCL-IDE as a reference for the different registers.

3.1 TMC5072-EVAL_UART.ino

```
/*
 * TMC5072-EVAL_UART.ino
 *
 * Created on: 20.02.2017
 *     Author: MN
 *
 *  The Trinamic TMC5072 motor controller and driver operates through Single
 *  Wire UART. This is 64 bits datagram (8 sync bytes, 8 bit slave address,
 *  8 bit register address and 32 bit word).
 *  Each register is specified by a one byte register address: with MSB = 0
 *  for read, MSB = 1 for write.
 *
 * Arduino Pins    Eval Board Pins
 * 18 TX1          40 SWIOP
 * 19 RX1          41 SWION
 *  2 DIO          38 DIO16 (SWSEL)
 *  4 DIO          33 SDO/RING
 *  3 DIO           8 DIO0 (DRV_ENN)
 * 10 DIO          23 CLK16
 *    GND           2 GND
 *    +5V           5 +5V
 */
```

©2017 TRINAMIC Motion Control GmbH & Co. KG, Hamburg, Germany
Terms of delivery and rights to technical change reserved.
Download newest version at www.trinamic.com
Read entire documentation.

```
#include "TMC5072_register.h"
#include "CRC.h"

#define CRC8_GEN 0x07

int SWSEL = 2;
int DRV_ENN = 3;
int SDO_RING = 4;
const int CLOCKOUT = 10;

void setup() {

  pinMode(SWSEL, OUTPUT);
  pinMode(CLOCKOUT,OUTPUT);
  pinMode(DRV_ENN, OUTPUT);
  pinMode(SDO_RING, OUTPUT);

  digitalWrite(DRV_ENN, LOW);
  //HIGH = power stage disabled, LOW = power stage enabled
  digitalWrite(SWSEL, HIGH);
  //HIGH = UART mode, LOW = SPI mode
  digitalWrite(SDO_RING, HIGH);
  //HIGH = ring mode, LOW = normal mode

  //setup timer2
  TCCR2A = ((1 << WGM21) | (1 << COM2A0));
  TCCR2B = (1 << CS20);
  TIMSK2 = 0;
  OCR2A = 0;

  Serial.begin(115200);
  Serial1.begin(115200);

  delay(500);
  uartWriteDatagram(0x00, TMC5072_SLAVECONF, 0x00000001); //SLAVEADDR to 1
  delay(500);

  tmc5072_initMotorDrivers();

}

void loop() {
  // put your main code here, to run repeatedly:

  uartRead(0x01, TMC5072_XACTUAL_1);  //read out XACTUAL of motor 1
  uartRead(0x01, TMC5072_XACTUAL_2);  //read out XACTUAL of motor 2

  uartWriteDatagram(0x01, TMC5072_XTARGET_1, 0x0007D000);
  //XTARGET=512000 | 10 revolutions with micro step = 256
  uartWriteDatagram(0x01, TMC5072_XTARGET_2, 0xFFF83000);
  //XTARGET=-512000 | 10 revolutions with micro step = 256

  delay(15000);
```

©2017 TRINAMIC Motion Control GmbH & Co. KG, Hamburg, Germany
Terms of delivery and rights to technical change reserved.
Download newest version at www.trinamic.com
Read entire documentation.

```
  uartRead(0x01, TMC5072_XACTUAL_1);  //read out XACTUAL of motor 1
  uartRead(0x01, TMC5072_XACTUAL_2);  //read out XACTUAL of motor 2

  uartWriteDatagram(0x01, TMC5072_XTARGET_1, 0x00000000); //XTARGET=0
  uartWriteDatagram(0x01, TMC5072_XTARGET_2, 0x00000000); //XTARGET=0

  delay(15000);
}

void uartWriteDatagram(uint8_t SLAVEADDR, uint8_t registerAddress,
      unsigned long datagram) {
  //TMC5072 takes 64 bit data: 8 sync + reserved, 8 chip address,
  //8 register address, 32 data, 8 CRC

  uint8_t CRC = 0;
  int temp;
  unsigned char buf[8];
  CRC = NextCRC(CRC, 0x05, CRC8_GEN);
  CRC = NextCRC(CRC, SLAVEADDR, CRC8_GEN);
  CRC = NextCRC(CRC, registerAddress|0x80, CRC8_GEN);
  CRC = NextCRC(CRC, (datagram >> 24) & 0xff, CRC8_GEN);
  CRC = NextCRC(CRC, (datagram >> 16) & 0xff, CRC8_GEN);
  CRC = NextCRC(CRC, (datagram >> 8) & 0xff, CRC8_GEN);
  CRC = NextCRC(CRC, datagram & 0xff, CRC8_GEN);

  buf[0] = 0x05;
  buf[1] = SLAVEADDR;
  buf[2] = registerAddress|0x80;
  buf[3] = (datagram >> 24) & 0xff;
  buf[4] = (datagram >> 16) & 0xff;
  buf[5] = (datagram >> 8) & 0xff;
  buf[6] = datagram & 0xff;
  buf[7] = CRC;

  temp = Serial1.write(buf, 8); //write datagram
  Serial1.flush();  //wait until all datas are written
  Serial1.readBytes(buf, 8); //clear input buffer
}

unsigned long uartRead(uint8_t SALVEADDR, uint8_t registerAddress) {

  uint8_t CRC = 0, temp;
  unsigned char buf[8];
  unsigned long dataBytes;

  CRC = NextCRC(CRC, 0x05, CRC8_GEN);
  CRC = NextCRC(CRC, SALVEADDR, CRC8_GEN);
  CRC = NextCRC(CRC, registerAddress, CRC8_GEN);
  buf[0] = 0x05;
  buf[1] = SALVEADDR;
  buf[2] = registerAddress;
  buf[3] = CRC;
  Serial1.write(buf, 4); //write datagram
```

©2017 TRINAMIC Motion Control GmbH & Co. KG, Hamburg, Germany
Terms of delivery and rights to technical change reserved.
Download newest version at www.trinamic.com
Read entire documentation.

```
  Serial1.flush();  //wait until all datas are written
  Serial1.readBytes(buf, 4); //clear input buffer

  Serial1.readBytes(buf, 8);
  temp = buf[2];
  dataBytes = buf[3]; //bit 32...24
  dataBytes <<= 8;
  dataBytes |= buf[4]; //bit 23...16
  dataBytes <<= 8;
  dataBytes |= buf[5]; //bit 15...8
  dataBytes <<= 8;
  dataBytes |= buf[6]; //bit 7...0

  CRC = 0;
  for(int i=0;i<7;i++)
  {
    CRC = NextCRC(CRC, buf[i], CRC8_GEN);
  }

  //show received bytes
  Serial.print("Received: 0x");
  for(int i=0;i<8;i++)
  {
    char tmp[16];
    sprintf(tmp, "%.2X", buf[i]);
    Serial.print(tmp);
  }
  Serial.print("\n");
  Serial.print("CRC: "); Serial.print(CRC,HEX);Serial.print(" <-> BUFFER: ");
  Serial.println(buf[7],HEX);

  return dataBytes;
}

void tmc5072_initMotorDrivers()
{
  //2-phase configuration Motor 1
  uartWriteDatagram(0x01, TMC5072_CHOPCONF_1, 0x00010135);
  uartWriteDatagram(0x01, TMC5072_IHOLD_IRUN_1, 0x00071400);

  //2-phase configuration Motor 2
  uartWriteDatagram(0x01, TMC5072_CHOPCONF_2, 0x00010135);
  uartWriteDatagram(0x01, TMC5072_IHOLD_IRUN_2, 0x00071400);


  //Reset positions
  uartWriteDatagram(0x01, TMC5072_RAMPMODE_1, TMC5072_MODE_POSITION);
  uartWriteDatagram(0x01, TMC5072_XTARGET_1, 0);
  uartWriteDatagram(0x01, TMC5072_XACTUAL_1, 0);
  uartWriteDatagram(0x01, TMC5072_RAMPMODE_2, TMC5072_MODE_POSITION);
  uartWriteDatagram(0x01, TMC5072_XTARGET_2, 0);
  uartWriteDatagram(0x01, TMC5072_XACTUAL_2, 0);

  //Standard values for speed and acceleration
```

©2017 TRINAMIC Motion Control GmbH & Co. KG, Hamburg, Germany
Terms of delivery and rights to technical change reserved.
Download newest version at www.trinamic.com
Read entire documentation.

```
  uartWriteDatagram(0x01, TMC5072_VSTART_1, 1);
  uartWriteDatagram(0x01, TMC5072_A1_1, 5000);
  uartWriteDatagram(0x01, TMC5072_V1_1, 26843);
  uartWriteDatagram(0x01, TMC5072_AMAX_1, 5000);
  uartWriteDatagram(0x01, TMC5072_VMAX_1, 100000);
  uartWriteDatagram(0x01, TMC5072_DMAX_1, 5000);
  uartWriteDatagram(0x01, TMC5072_D1_1, 5000);
  uartWriteDatagram(0x01, TMC5072_VSTOP_1, 10);

  uartWriteDatagram(0x01, TMC5072_VSTART_2, 1);
  uartWriteDatagram(0x01, TMC5072_A1_2, 5000);
  uartWriteDatagram(0x01, TMC5072_V1_2, 26843);
  uartWriteDatagram(0x01, TMC5072_AMAX_2, 5000);
  uartWriteDatagram(0x01, TMC5072_VMAX_2, 100000);
  uartWriteDatagram(0x01, TMC5072_DMAX_2, 5000);
  uartWriteDatagram(0x01, TMC5072_D1_2, 5000);
  uartWriteDatagram(0x01, TMC5072_VSTOP_2, 10);
}
```

3.2 TMC5072_register.h

```
/*
 * TMC5072-EVAL_UART.ino
 *
 * Created on: 20.02.2017
 *     Author: MN
 */

#ifndef TMC5072_REGISTER_H
#define TMC5072_REGISTER_H

   // ===== TMC5072 register set =====

   #define TMC5072_GCONF        0x00
   #define TMC5072_GSTAT        0x01
   #define TMC5072_IFCNT        0x02
   #define TMC5072_SLAVECONF    0x03
   #define TMC5072_INP_OUT      0x04
   #define TMC5072_X_COMPARE    0x05

   #define TMC5072_PWMCONF_1    0x10
   #define TMC5072_PWM_STATUS_1 0x11

   #define TMC5072_PWMCONF_2    0x18
   #define TMC5072_PWM_STATUS_2 0x19

   #define TMC5072_RAMPMODE_1   0x20
   #define TMC5072_XACTUAL_1    0x21
   #define TMC5072_VACTUAL_1    0x22
   #define TMC5072_VSTART_1     0x23
   #define TMC5072_A1_1         0x24
   #define TMC5072_V1_1         0x25
   #define TMC5072_AMAX_1       0x26
   #define TMC5072_VMAX_1       0x27
```

©2017 TRINAMIC Motion Control GmbH & Co. KG, Hamburg, Germany
Terms of delivery and rights to technical change reserved.
Download newest version at www.trinamic.com
Read entire documentation.

```
#define TMC5072_DMAX_1        0x28
#define TMC5072_D1_1          0x2A
#define TMC5072_VSTOP_1       0x2B
#define TMC5072_TZEROWAIT_1   0x2C
#define TMC5072_XTARGET_1     0x2D
#define TMC5072_IHOLD_IRUN_1  0x30
#define TMC5072_VCOOLTHRS_1   0x31
#define TMC5072_VHIGH_1       0x32
#define TMC5072_VDCMIN_1      0x33
#define TMC5072_SWMODE_1      0x34
#define TMC5072_RAMPSTAT_1    0x35
#define TMC5072_XLATCH_1      0x36
#define TMC5072_ENCMODE_1     0x38
#define TMC5072_XENC_1        0x39
#define TMC5072_ENC_CONST_1   0x3A
#define TMC5072_ENC_STATUS_1  0x3B
#define TMC5072_ENC_LATCH_1   0x3C

#define TMC5072_RAMPMODE_2    0x40
#define TMC5072_XACTUAL_2     0x41
#define TMC5072_VACTUAL_2     0x42
#define TMC5072_VSTART_2      0x43
#define TMC5072_A1_2          0x44
#define TMC5072_V1_2          0x45
#define TMC5072_AMAX_2        0x46
#define TMC5072_VMAX_2        0x47
#define TMC5072_DMAX_2        0x48
#define TMC5072_D1_2          0x4A
#define TMC5072_VSTOP_2       0x4B
#define TMC5072_TZEROWAIT_2   0x4C
#define TMC5072_XTARGET_2     0x4D
#define TMC5072_IHOLD_IRUN_2  0x50
#define TMC5072_VCOOLTHRS_2   0x51
#define TMC5072_VHIGH_2       0x52
#define TMC5072_VDCMIN_2      0x53
#define TMC5072_SWMODE_2      0x54
#define TMC5072_RAMPSTAT_2    0x55
#define TMC5072_XLATCH_2      0x56
#define TMC5072_ENCMODE_2     0x58
#define TMC5072_XENC_2        0x59
#define TMC5072_ENC_CONST_2   0x5A
#define TMC5072_ENC_STATUS_2  0x5B
#define TMC5072_ENC_LATCH_2   0x5C

#define TMC5072_MSLUT0        0x60
#define TMC5072_MSLUT1        0x61
#define TMC5072_MSLUT2        0x62
#define TMC5072_MSLUT3        0x63
#define TMC5072_MSLUT4        0x64
#define TMC5072_MSLUT5        0x65
#define TMC5072_MSLUT6        0x66
#define TMC5072_MSLUT7        0x67
#define TMC5072_MSLUTSEL      0x68
#define TMC5072_MSLUTSTART    0x69
```

©2017 TRINAMIC Motion Control GmbH & Co. KG, Hamburg, Germany
Terms of delivery and rights to technical change reserved.
Download newest version at www.trinamic.com
Read entire documentation.

```
#define TMC5072_MSCNT_1       0x6A
#define TMC5072_MSCURACT_1    0x6B
#define TMC5072_CHOPCONF_1    0x6C
#define TMC5072_COOLCONF_1    0x6D
#define TMC5072_DCCTRL_1      0x6E
#define TMC5072_DRVSTATUS_1   0x6F

#define TMC5072_MSCNT_2       0x7A
#define TMC5072_MSCURACT_2    0x7B
#define TMC5072_CHOPCONF_2    0x7C
#define TMC5072_COOLCONF_2    0x7D
#define TMC5072_DCCTRL_2      0x7E
#define TMC5072_DRVSTATUS_2   0x7F

#define TMC5072_PWMCONF_1     0x10
#define TMC5072_PWM_STATUS    0x11

#define TMC5072_RAMPMODE      0x00
#define TMC5072_XACTUAL       0x01
#define TMC5072_VACTUAL       0x02
#define TMC5072_VSTART        0x03
#define TMC5072_A1            0x04
#define TMC5072_V1            0x05
#define TMC5072_AMAX          0x06
#define TMC5072_VMAX          0x07
#define TMC5072_DMAX          0x08
#define TMC5072_D1            0x0A
#define TMC5072_VSTOP         0x0B
#define TMC5072_TZEROWAIT     0x0C
#define TMC5072_XTARGET       0x0D
#define TMC5072_IHOLD_IRUN    0x10
#define TMC5072_VCOOLTHRS     0x11
#define TMC5072_VHIGH         0x12
#define TMC5072_VDCMIN        0x13
#define TMC5072_SWMODE        0x14
#define TMC5072_RAMPSTAT      0x15
#define TMC5072_XLATCH        0x16
#define TMC5072_ENCMODE       0x18
#define TMC5072_XENC          0x19
#define TMC5072_ENC_CONST     0x1A
#define TMC5072_ENC_STATUS    0x1B
#define TMC5072_ENC_LATCH     0x1C
#define TMC5072_CHOPCONF      0x6C
#define TMC5072_COOLCONF      0x6D
#define TMC5072_DRVSTATUS     0x6F

//Motorbits und Write-Bit
#define TMC5072_MOTOR0        0x20
#define TMC5072_MOTOR1        0x40
#define TMC5072_WRITE         0x80

//Rampenmodi (Register TMC5072_RAMPMODE)
#define TMC5072_MODE_POSITION   0
```

©2017 TRINAMIC Motion Control GmbH & Co. KG, Hamburg, Germany
Terms of delivery and rights to technical change reserved.
Download newest version at www.trinamic.com
Read entire documentation.

```
  #define TMC5072_MODE_VELPOS     1
  #define TMC5072_MODE_VELNEG     2
  #define TMC5072_MODE_HOLD       3

  //Endschaltermodusbits (Register TMC5072_SWMODE)
  #define TMC5072_SW_STOPL_ENABLE   0x0001
  #define TMC5072_SW_STOPR_ENABLE   0x0002
  #define TMC5072_SW_STOPL_POLARITY 0x0004
  #define TMC5072_SW_STOPR_POLARITY 0x0008
  #define TMC5072_SW_SWAP_LR        0x0010
  #define TMC5072_SW_LATCH_L_ACT    0x0020
  #define TMC5072_SW_LATCH_L_INACT  0x0040
  #define TMC5072_SW_LATCH_R_ACT    0x0080
  #define TMC5072_SW_LATCH_R_INACT  0x0100
  #define TMC5072_SW_LATCH_ENC      0x0200
  #define TMC5072_SW_SG_STOP        0x0400
  #define TMC5072_SW_SOFTSTOP       0x0800

  //Statusbitss (Register TMC5072_RAMPSTAT)
  #define TMC5072_RS_STOPL          0x0001
  #define TMC5072_RS_STOPR          0x0002
  #define TMC5072_RS_LATCHL         0x0004
  #define TMC5072_RS_LATCHR         0x0008
  #define TMC5072_RS_EV_STOPL       0x0010
  #define TMC5072_RS_EV_STOPR       0x0020
  #define TMC5072_RS_EV_STOP_SG     0x0040
  #define TMC5072_RS_EV_POSREACHED  0x0080
  #define TMC5072_RS_VELREACHED     0x0100
  #define TMC5072_RS_POSREACHED     0x0200
  #define TMC5072_RS_VZERO          0x0400
  #define TMC5072_RS_ZEROWAIT       0x0800
  #define TMC5072_RS_SECONDMOVE     0x1000
  #define TMC5072_RS_SG             0x2000

  #define MOTOR_ADDR(m) (0x20 << m )
  #define MOTOR_ADDR_DRV(m)  (m << 4)

#endif
```

3.3 TMC5072_register.h

```
/*************************************************
  Projekt:  TMC5130 and TMC5072 CRC calculation

  Modul:    CRC.h
            CRC-Calculation for UART interfacing

  Hinweise: Start with CRC-Register=0,
            then call NextCRC for each byte to be sent
            or each by received. Send CRC byte last or
            check received CRC

  Datum:    14.6.2011 OK
*************************************************/
```

©2017 TRINAMIC Motion Control GmbH & Co. KG, Hamburg, Germany
Terms of delivery and rights to technical change reserved.
Download newest version at www.trinamic.com
Read entire documentation.

```
#ifndef CRC_H
#define CRC_H

  #include "TypeDefs.h"

  uint8 NextCRCSingle(uint8 Crc, uint8 Data, uint8 Gen, uint8 Bit);
  uint8 NextCRC(uint8 Crc, uint8 Data, uint8 Gen);

#endif
```

3.4 TMC5072_register.h

```
/*****************************************************
  Projekt:  TMC5130 and TMC5072 CRC calculation

  Modul:    CRC.cpp
            CRC-Calculation for UART interfacing

  Hinweise: Start with CRC-Register=0,
            then call NextCRC for each byte to be sent
            or each by received. Send CRC byte last or
            check received CRC

  Datum:    14.6.2011 OK

  GeÃ¤ndert: crc counting direction, 17.06.2011, LL, SK
*****************************************************/

#include "CRC.h"


uint8 NextCRCSingle(uint8 Crc, uint8 Data, uint8 Gen, uint8 Bit)
{
  uint8 Compare;

  Compare=Data<<(7-Bit);
  Compare&=0x80;

  if((Crc & 0x80) ^ (Compare))
    return (Crc << 1) ^ Gen;
  else
    return (Crc << 1);
}

uint8 NextCRC(uint8 Crc, uint8 Data, uint8 Gen)
{
  uint8 Temp;
  int i;

  Temp=Crc;
  for(i=0; i<=7; i++)
  {
    Temp=NextCRCSingle(Temp, Data, Gen, i);
  }
```

©2017 TRINAMIC Motion Control GmbH & Co. KG, Hamburg, Germany
Terms of delivery and rights to technical change reserved.
Download newest version at www.trinamic.com
Read entire documentation.

```
  return Temp;
}
```

3.5 TMC5072_register.h

```
/*
 * Types.h
 *
 *  Created on: 29.09.2016
 *      Author: ed
 */

#ifndef API_TYPEDEFS_H
#define API_TYPEDEFS_H

   // unsigned types
   typedef unsigned char         u8;
   typedef unsigned short        u16;
// typedef unsigned int          u32;
   typedef unsigned long long       u64;

   typedef unsigned char           uint8;
   typedef unsigned short int      uint16;
   typedef unsigned long int       uint32;
   typedef unsigned long long       uint64;

   #define u8_MAX             (u8)255
   #define u10_MAX               (u16)1023
   #define u12_MAX               (u16)4095
   #define u15_MAX               (u16)32767
   #define u16_MAX               (u16)65535

   #define u20_MAX               (u32)1048575uL
   #define u24_MAX               (u32)16777215uL
   #define u32_MAX               (u32)4294967295uL

   // signed types
   typedef signed char        s8;
   typedef signed short          s16;
   typedef signed int            s32;
   typedef signed long long      s64;

   typedef signed char             int8;
   typedef signed short int        int16;
   typedef signed long int         int32;
   typedef signed long long      int64;

   #define s16_MAX               (s16)32767
   #define s16_MIN               (s16)-32768
   #define s24_MAX               (s32)8388607
   #define s24_MIN               (s32)-8388608
   #define s32_MAX               (s32)2147483647
   #define s32_MIN               (s32)-2147483648
```

©2017 TRINAMIC Motion Control GmbH & Co. KG, Hamburg, Germany
Terms of delivery and rights to technical change reserved.
Download newest version at www.trinamic.com
Read entire documentation.

```
   #define FALSE                  0
   #define TRUE                   1

   // parameter access
   #define READ                0
   #define WRITE               1

   // "macros"
   #define MIN(a,b) (a<b) ? (a) : (b)
   #define MAX(a,b) (a>b) ? (a) : (b)

   #ifndef UNUSED
      #define UNUSED(x) (void)(x)
   #endif

   #ifndef NULL
      #define NULL 0x00u
   #endif

#endif /* API_TYPEDEFS_H */
```

©2017 TRINAMIC Motion Control GmbH & Co. KG, Hamburg, Germany
Terms of delivery and rights to technical change reserved.
Download newest version at www.trinamic.com
Read entire documentation.

4 Revision History

Version	Date	Author	Description
V0.10	2017-Feb-20	MN	Initial Version

Table 1: Document Revision

©2017 TRINAMIC Motion Control GmbH & Co. KG, Hamburg, Germany
Terms of delivery and rights to technical change reserved.
Download newest version at www.trinamic.com
Read entire documentation.