Ert Ülle

Design of CAN-node

Metropolia University of Applied Sciences

Bachelor of Engineering

Electrical engineering

Thesis

7 March 2018

Author Title Number of Pages Date	Ert Ülle Design of CAN-node 18 pages + 2 appendices 7 March 2018
Degree	Bachelor of Engineering
Degree Programme	Electrical engineering
Professional Major	Electronics
Instructors	Senior Lecturer Vesa Linja-aho

This study was done to the Metropolia Motorsport RDI project. The goal was to design a small, easy to use, affordable and CAN capable I/O PCB prototype. Application of this PCB is to bring sensor data gathered from race-car to CAN-bus and to control peripheral devices based on the commands received through the bus.

The schematic and printed circuit board was designed using the PADS-software.

Main guidelines while choosing the components for the PCB were cost effectiveness and ease of use. Because of this, main components of the board are widely used in the industry and well documented. The chosen components include ATSAMD21G18-microcontroller, MCP2515-CAN controller and TCAN332-CAN transceiver.

Quick testing proved the first prototype to be working. For the next iteration, size of the PCB could still be reduced, needed I/O functions adjusted, proper casing designed and suitable connector chosen to ease integrating the PCB into a vehicle.

Keywords	CAN, Controller Area Network, Microcontroller, GPIO

Tekijä Otsikko Sivumäärä Aika	Ert Ülle Can-solmun suunnittelu 18 sivua + 2 liitettä 7.3.2018
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Sähkötekniikka
Ammatillinen pääaine	Elektroniikka
Ohjaajat	Lehtori Vesa Linja-aho

Tämä insinöörityö on tehty Metropolia Motorsportin TKI-projektille. Insinöörityön tavoitteena oli suunnitella helppokäyttöinen, pienikokoinen ja edullinen I/O-piirikortin prototyyppi CAN-väylään. Piirikortin käyttötarkoituksena on tuoda kilpa-auton antureista luettua tietoa CAN-väylään sekä ohjata auton oheislaitteita väylän kautta saatujen käskyjen perusteella.

Piirilevy suunniteltiin PADS-ohjelmistoa käyttäen.

Komponenttien valinnassa keskityttiin kustannustehokkuuteen ja helppokäyttöisyyteen. Tästä johtuen valittiin mahdollisimman yleisiä komponentteja, joihin löytyy kattavasti dokumentaatiota. Piirilevyn pääkomponenteiksi valittiin ATSAMD21G18-mikrokontrolleri, MCP2515-CAN ohjain sekä TCAN332-CAN lähetin-vastaanotin.

Ensimmäinen prototyyppi todettiin toimivaksi lyhyillä testauksilla. Seuraavaan versioon voidaan pienentää piirilevyn kokoa entisestään, tarkistella tarvittavia I/O-ominaisuuksia, valita kotelointi sekä käyttötarkoitukseen sopiva liitin autoon integroimisen helpottamiseksi.

Avainsanat	CAN, Controller Area Network, Mikrokontrolleri, Mikro-ohjain

Metropolia

Contents

List of Abbreviations

EMI	Electromagnetic interference
ECU	Engine Control Unit
SAE	Society of Automotive Engineers
PCB	Printed Circuit Board
CAN	Controller Area Network
I/O	Input/Output
ADC	Analog-to-digital converter
MCU	Microcontroller
SPI	Serial Peripheral Interface
SOIC	Small Outline Integrated Circuit

1 Introduction

Electric race-cars use an increasing amount of sensor data for dynamic vehicle control and data acquisition for later analysis. This leads to several challenges that include complex and heavy wiring harness, as well as long sensor wires that cause reliability problems with read sensor data because of EMI Electromagnetic interference. To fight these problems, the idea of using multiple small modules was born. Reading the sensors on spot and sending values trough digital communication bus to ECU engine control unit or logger allows us to use short analog cables, which improves the reliability of the read signal. Using the digital bus also allows us to send data from several sensors using only few wires. This helps to simplify and reduce weight of the wiring harness.

Going through commercially available products showed that there does not appear to be valid options for our use. All of the units either have really high cost or the physical parameters as size and weight exceed any reasonable limits for a small lightweight race-car (see chapter 3) [1;2;3].

After researching available options, it was decided to try developing an own solution. As design guidelines for the new board, low cost and small size were chosen, as there were no products found to fulfill these requirements. Simplicity was also taken into account so the device would have as mild learning curve as possible on both hardware and software. This allows other team members to use and modify device properties without spending too much time.

2 Metropolia Motorsport

2.1 Formula Student

The first formula student competition was held in 1981 at University of Texas campus by Formula SAE Society of Automotive Engineers and the concept has been gaining popularity ever since. The whole idea of Formula Student is more of an engineering contest than racing, as the drivers are students themselves, in other words amateurs. There are around 550 combustion teams and 150 electric teams competing in this series today [4].

2.2 Metropolia Motorsport

Metropolia Motorsport is a research, development and innovation project of Metropolia University of Applied Sciences. The aim of the project is to design and build a formula type race-car (Figure 1) every year which is used to compete in Formula Student series against other universities from all over the world. Metropolia Motorsport was founded in year 2000 and first car was built in 2002. In 2012 the team decided to change for electric vehicle series and first electric car was built in 2013.

Figure 1. HPF017 car

Overall 14 cars have been designed and manufactured so far by Metropolia Motorsport. The whole project is managed and carried out by students. Project is partly funded by the university but a huge part of funding is self-acquired through sponsors [5].

2.3 Competitions

Competitions consist of multiple static and dynamic events (Table 1).

Table 1. All the static and dynamic events and their respective maximum score.

EVENT	Max. Score
Endurance	325
Efficiency	100
Autocross	100
Skid Pad	75
Acceleration	75
Design	150
Cost	100
Business Plan	75

Additionally one extremely important part is scrutineering, as the car has to be designed and manufactured in a very limited schedule, it is a real challenge to get everything together and rules compliant in time. There are strict requirements for structural integrity and safety systems of the car written in the rules which must be fulfilled. Failing to do so will result in not being allowed to participate in dynamic events [6].

3 CAN Node

CAN Controller Area Network node is a small microcontroller board designed to operate as an interface between various sensors, peripherals and CAN bus onboard our electric formula. Despite of node being designed for our race car it could be used within several systems that utilize can bus, as it is able to further expand I/O Input/output functionalities of various CAN based control units or logging devices.

Commercially available products were investigated to get some direction of the measurement parameters and features generally offered in those. Most resembling products by their general idea that had enough comprehensive information available to make any kind of analysis were MoTeC SVIM [2] and McLaren SN-320 [3] (Table 2).

Table 2. General parameters of commercial nodes

	SVIM	SN-320
Resolution	**12-15-bit 0-5V**	**12-bit 0-5V**
Sample rate	**Up to 5000Hz**	**Up to 10000Hz**
General	**1 x CAN 24 x Analog inputs Supply: 7-30V Size: 48x90x26.2mm Weight: 150g Price: 2000€+ Case: Anodized aluminum**	**1 x CAN 24 x Analog inputs Supply: 7.5-16V Size: 110x72x24mm Weight:<150g Price: Not Found Case: Painted magnesium alloy**

Examining the data acquired from professional products shows that the technical parameters of the measurements are somewhat conservative, as the maximum resolution is 15-bit and maximum sample rate is 10 kHz, while most inputs are running with 12-bit ADC Analog-to-digital converter with 1 kHz sample rate. Apparently these parameters are considered sufficient for automotive use, as much more powerful hardware has been available on the market for quite some time but have not made their way into race car data acquisition or control applications.

Based on the values acquired about commercial products and from personal experience it was concluded that a somewhat fast 12-bit ADC should be sufficient for measuring most of the race car related values. If higher resolution is necessary, it can be achieved by oversampling. However, that will affect the maximum sample rate directly.

In this application also a wide input voltage range is an important aspect as the car has 5 V, 12 V and 19.2-25.2 V extra low voltage systems. It simplifies the installation drastically, if the technician is able to utilize whichever supply is available. For the same reason the node should be able to supply most of the sensors directly, if necessary. This leads to certain requirements related to implementation of voltage regulation on the node and a common solution of using low-dropout linear regulator is not best suited in this case, as it has low efficiency and will cause problems with heat at higher input voltages and supply currents.

One important feature that might be easily neglected during hardware design is overall usability of the product. Commercial products have highly developed graphical user interfaces available that simplify configuring the node. This drastically reduces time needed for configuration and smoothens the learning curve for a new, inexperienced user. In this case making a polished interface is somewhat out of the question as the resources are limited and to include all the necessary features is a huge project. This leads to the fact that most versatile hardware must be chosen, considering the selection of programming environments available for different purposes and skill levels. The aim was to have at least some popular programming tools supported including for example Atmel Studio, Arduino, MATLAB/Simulink and LabVIEW.

As designing a proper casing can take a lot of time, increase size of the product and end up expensive, it will be neglected at this prototyping stage. The board can still be protected by potting it with some epoxy or hot glue.

4 Main Components

4.1 Microchip ATSAMD21G18A

As the main microcontroller it was chosen to use 32-bit Cortex-M0+-based 48 MHz ATSAMD21G18 [7]. Low price and high performance against many other 8-bit and 32-bit units was a decisive factor for choosing this unit. Most of the cheaper options had too weak performance on the analog-to-digital converter side, which is one of the most important aspects for reading sensor data. This 48 pin MCU Microcontroller was picked in Thin Quad Flat Package (Figure 2) instead of Quad-flat No-leads or Wafer Level Chip Scale Package, as this way it does not require any special tools for assembly; basic soldering iron is enough.

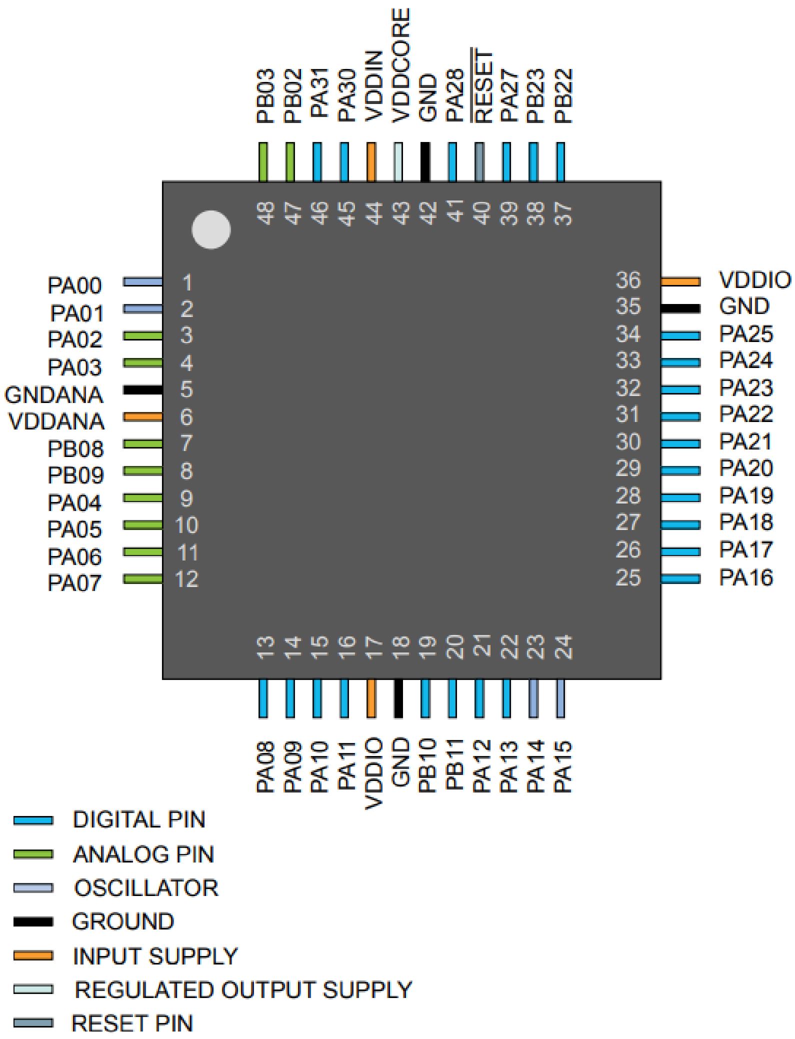


Figure 2. ATSAMD21G18 Thin Quad Flat Package pinout

MCU has 38 general-purpose input/output pins, ten of which can be configured to work as digital inputs or outputs, analog inputs and pulse width modulation outputs, whichever is needed. Built in analog-to-digital converter is a 12-bit unit capable of reading up to 350 000 samples per second. By oversampling 16-bit resolution can be achieved. Also common I²C, SPI Serial Peripheral Interface, USB and UART connections are available.

The same MCU is also used in Arduino M0 [8], Genuino Zero and few other development boards. This should give the ability to use several different integrated development environments, such as Mbed, Atmel Studio or Arduino IDE. This also makes a huge assortment of libraries available for different hardware and software implementations. Arduino devices also support using MATLAB [9] and Simulink for programming, simulation and debugging. These are widely used in motorsports and other industries and, as such, are a useful addition to available tools.

4.2 Microchip MCP2515

This is one of the most commonly used stand-alone CAN controllers. Because of its cheap price and good availability it is also widely spread among hobby users. For communicating with host microcontroller it is using Serial Peripheral Interface. As a package type, it was chosen to use 18-Pin SOIC Small Outline Integrated Circuit for easier hand assembly. Being commonly used, provides user with multiple readily available libraries for interfacing with this particular controller [10].

4.3 Texas Instruments TCAN332

CAN controller also requires a transceiver to be able to communicate through the bus. For this the most appropriate component seemed to be TCAN332DR. It was one of the cheapest and well available transceivers with the correct supply voltage range. Absence additional pins or features make it easy to use. As a package 8-pin SOIC was chosen [11].

4.4 ON Semiconductor NCV8402D

For controlling external peripherals, some type of driving circuit would have to be used to withstand higher loads. For this application, NCV8402D smart dual low side driver seemed to be the right solution. Despite the low price and small size it is able to drive loads up to 42-volts and 2-amperes. Furthermore it has built in overcurrent, overvoltage, overtemperature and electrostatic discharge protection. All this protection allows not to worry about device breaking down, if being mishandled by mistake [12].

4.5 XP Power TR05S3V3

As for the power supply for the whole circuit, it was decided to use a switching regulator. Although it is somewhat huge compared to other components (Figure 3) TR05S3V3 fulfills all the necessary requirements for this particular case.

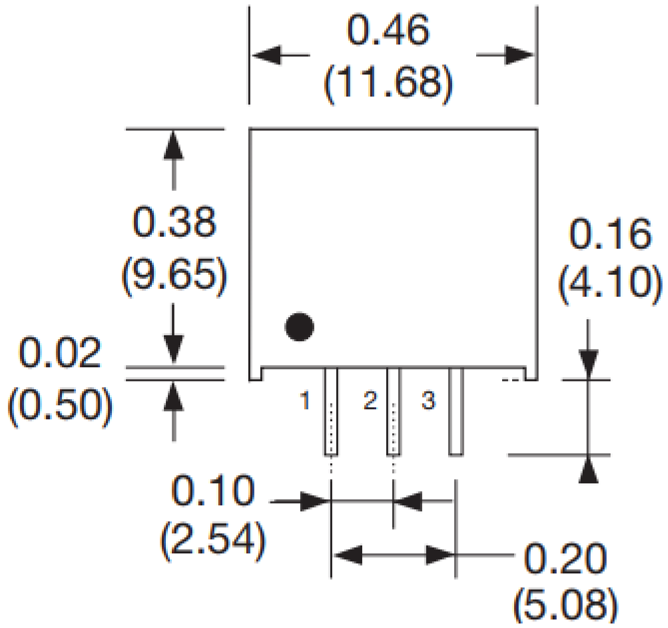


Figure 3. Mechanical dimensions of the switching regulator

It has a wide input range so there is no need to worry about input voltage, high efficiency helps to generate less heat, price of this component is relatively low and up to 2-watt power rating allows to supply any necessary sensors directly from the board. This regulator also features a built in short circuit protection [13].

5 Design

5.1 Schematic

Schematic and printed circuit board were designed by using PADS software. While designing the schematic the aim was to keep it simple and readable. At this point also some additional components were added to support the main parts, full bill of materials in Table 3.

Table 3. Bill of materials

Index	Quantity	Part Number	Description	Unit Price USD
1	2	1727-5217-1-ND	DIODE SCHOTTKY 60V 2A SOD123W	0.38
2	1	296-43711-1-ND	IC TXRX CAN 3.3V 8SOIC	2.11
3	1	1470-2257-5-ND	DC/DC CONVERTER 3.3V 2W	3.59
4	1	MCP2515T-I/SOCT-ND	IC CAN CONTROLLER W/SPI 18SOIC	1.91
5	1	NCV8402ADDR2GOSCT-ND	IC DVR LOW SIDE 8-SOIC	1.02
6	1	497-13262-1-ND	TVS DIODE 24VWM 40VC SOT23-3L	0.35
7	1	ATSAMD21G18A-AUTCT-ND	IC MCU 32BIT 256KB FLASH 48TQFP	3.29
8	1	644-1313-1-ND	CRYSTAL 32.768KHZ 12.5PF SMD	0.94
9	1	644-1049-1-ND	CRYSTAL 16.0000MHZ 8PF SMD	0.75
10	4	399-1281-1-ND	CAP CER 0.1UF 25V X7R 0603	0.03
11	2	399-8979-1-ND	CAP CER 10PF 25V C0G/NP0 0603	0.10
12	2	399-9004-1-ND	CAP CER 15PF 16V C0G/NP0 0603	0.10
13	3	1276-1946-1-ND	CAP CER 1UF 10V X7R 0603	0.10
14	1	1276-1098-1-ND	CAP CER 4700PF 50V X7R 0603	0.01
15	4	311-10KGRCT-ND	RES SMD 10K OHM 5% 1/10W 0603	0.01
16	2	311-60.4HRCT-ND	RES SMD 60.4 OHM 1% 1/10W 0603	0.02
				Total including quantity
				16.1

ESDCAN24 [14] transient voltage suppressor was added to protect the CAN transceiver from ESD, there is also a cap attached to termination resistors to form a low pass filter for noise reduction. Some 10 kilo-ohm resistors were used as pull-up or pull-down where necessary (Figure 4) (Appendix 1).

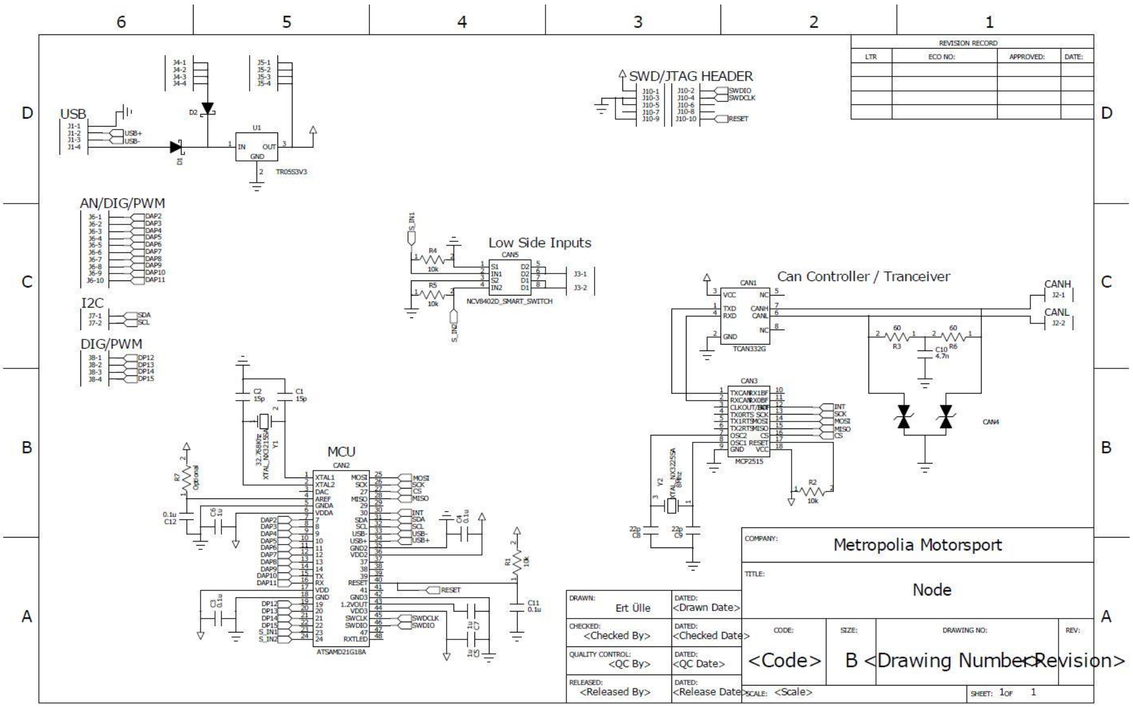


Figure 4. Schematic of the CAN-node.

Additionally some external capacitors were added for the crystal oscillators, values of which were calculated with a formula provided by Microchip (Figure 5).

37.12.1.2 Crystal Oscillator Characteristics

The following table describes the characteristics for the oscillator when a crystal is connected between XIN and XOUT . The user must choose a crystal oscillator where the crystal load capacitance C_L is within the range given in the table. The exact value of C_L can be found in the crystal datasheet. The capacitance of the external capacitors (C_{LEXT}) can then be computed as follows:

$$C_{LEXT} = 2(C_L + - C_{STRAY} - C_{SHUNT})$$

where C_{STRAY} is the capacitance of the pins and PCB, C_{SHUNT} is the shunt capacitance of the crystal.

$$2(12.5\ \mathrm{pF} - 4\ \mathrm{pF} - 1\ \mathrm{pF}) = 15\ \mathrm{pF}$$

Figure 5. Formula provided by Microchip [15].

5.2 Printed Circuit Board

During the design of the PCB Printed Circuit Board focus was kept on manufacturability and ease of testing, as this is a first prototype. As material it was chosen to use FR4, as it is the cheapest option available at PCB suppliers and is sufficient for this application. Most of the components have some space left around them so they are easier to assemble and measure if needed (Figure 6) (Appendix 2).

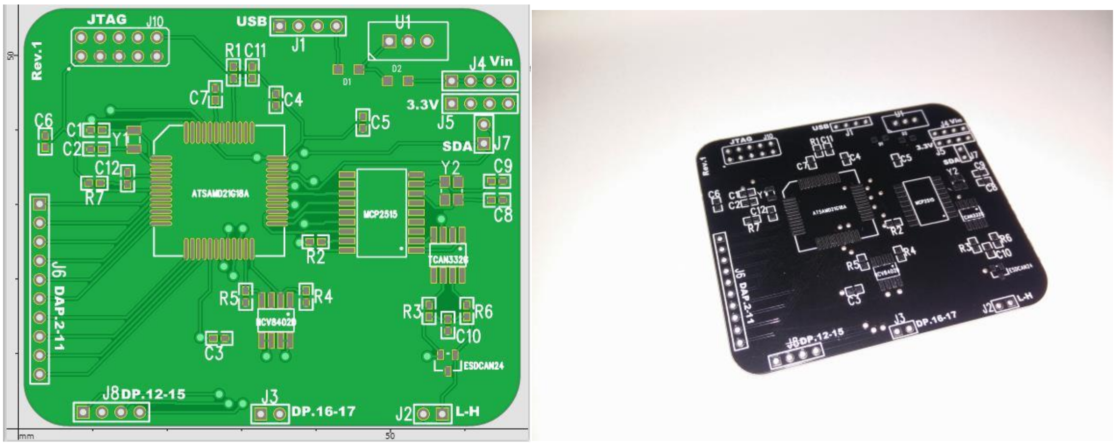


Figure 6. Rendered and actual top side of the PCB

Still all the communication traces have been kept short to reduce any interference. Furthermore, it has also a ground plane on the top side of the PCB. Bottom is used as a supply plane and all the supply pins are brought with vias to the top side where necessary. As the supplier for the PCBs Seeed Fusion PCB [16] was used, as it is one of the cheapest and previous experiences have proven their quality to be quite good.

6 Software

6.1 CAN Bus

CAN-bus was originally designed for data transfer between multiple in-vehicle control units in 1985 by Bosch. Afterwards it has widely spread in many other products such as machining and healthcare equipment. Couple advantages of the CAN are light and affordable construction, as the hardware is cheap and as data transfer wire it uses a simple twisted pair (Figure 7).

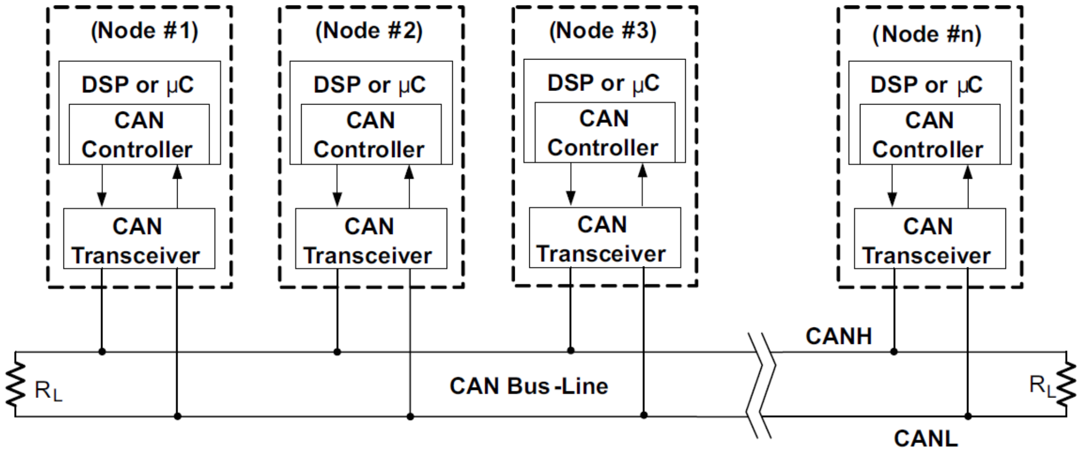


Figure 7. Basic structure of CAN network

Broadcast communication also simplifies the use, as every node of the network is able to filter the data it requires. Perhaps the biggest advantage is a high fault-tolerance, as CAN is a differential bus it is extremely resilient to any type of noise [17].

6.2 Arduino IDE

Arduino integrated development environment (IDE) [18] is a free package available from Arduino website for easy development of embedded systems. It contains a simple code editor, compiler and a set of tools for uploading code to microcontroller. The programming language itself is based on C++ but simplified on the hardware setup side (Listing 1).

```
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off
  delay(1000);                       // wait for a second
}
```

Listing 1. Full code required for blinking a LED using 928 bytes of FLASH and 9 bytes of SRAM

It was chosen to use Arduino for this project, as it helps other team members by being simple to learn for new users.

6.3 Testing

As a first test the aim was find out how fast can the ADC work with default Arduino configuration files. Test was conducted by taking timestamp before and after reading the inputs and reducing those values from each other (Listing 2). Default values turned out to be 400 microseconds for a single input and 2500 microseconds for six inputs. This converts into 2500 reads/second and 400 reads/second, respectively. By adjusting the prescaler for analog read command, it was possible to increase the speed to 16129 reads/second for a single input and 2777 reads/second for six. Further increase of sample rate was possible, but it caused instability in ADC performance. The sample rate started to float between 58823-76923 reads/second.

```
int sensor1 = 0;
int sensor2 = 0;
int sensor3 = 0;
int sensor4 = 0;
int sensor5 = 0;
int sensor6 = 0;

void setup() {

int16_t ctrlb = 0x0400; // set prescaler
  ADC->CTRLB.reg =  ctrlb; // set prescaler
    pinMode(LED_BUILTIN, OUTPUT); //set led pin
     SerialUSB.begin(9600);  // initialize serial communications at 9600 bps:
}
void loop() {
  // read the analog in value:
  uint32_t ts1 = micros(); //interval start
    sensor1 = analogRead(A0);
/*    sensor2 = analogRead(A1);
     sensor3 = analogRead(A2);
     sensor4 = analogRead(A3);
     sensor5 = analogRead(A4);
     sensor6 = analogRead(A5);*/
        uint32_t ts2 = micros(); //interval stop

    digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (for debugging)
  delay(200);
  SerialUSB.print("sensor = ");   // print the values to the serial monitor
  SerialUSB.println(sensor1);
/*  SerialUSB.println(sensor2);
  SerialUSB.println(sensor3);
  SerialUSB.println(sensor4);
  SerialUSB.println(sensor5);
  SerialUSB.println(sensor6);*/

  // print the time interval in microseconds
  SerialUSB.print(ts2-ts1);
  SerialUSB.println("micros");
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off (for debugging)
  delay(200);
}
```

Listing 2. ADC speed test code

The second part of the testing was concentrated on testing the SPI and CAN-controller performance. Logic of the test was similar to previous one but the timing was performed for the CAN-message send (Listing 3) and receive commands. As a reference, Arduino nano was used for performance comparison. With default settings a transfer rate of 465 messages/second was achieved, after adjusting the SPI frequency transfer rate increased to around 650 messages/second. Further tweaks of any related parameters did not affect the performance significantly. This was somewhat disappointing, as Arduino nano with somewhat the same code and weaker hardware was able to perform at 5435 messages/second. Poor performance is apparently related to how SERCOM communication interface is implemented in the Arduino configuration files. The user is able to setup multiple communication buses to whichever pins he wants but it uses multiple libraries for the setup of the communication pins that it runs through during every loop. This causes unnecessary delays for communication.

```
#include <mcp_can.h>
#include <SPI.h>
#define Serial SerialUSB
/*#include "wiring_private.h" // pinPeripheral() function
#define SPI mySPI
SPIClass mySPI (&sercom1, 12, 13, 11, SPI_PAD_0_SCK_1, SERCOM_RX_PAD_3);
*/
const int SPI_CS_PIN = 9;
int interval;
int ts1;
int ts2;

MCP_CAN CAN(SPI_CS_PIN);                                    // Set CS pin

void setup()
{
    Serial.begin(115200);
/*    mySPI.begin();

  pinPeripheral(11, PIO_SERCOM);
  pinPeripheral(12, PIO_SERCOM);
  pinPeripheral(13, PIO_SERCOM);
*/
START_INIT:

    if(CAN_OK == CAN.begin(CAN_500KBPS))                   // init can bus :
baudrate = 500k
    {
        Serial.println("CAN BUS initialization successful");
    }
    else
    {
        Serial.println("CAN BUS initialization failed");
        Serial.println("Try again");
        delay(100);
        goto START_INIT;
    }
}


void loop()
```

```
{
    // send data:  id = 0x00, standrad frame, data len = 8, msg: data buf
    ts1 = micros(); //interval start
    unsigned char msg[8] = { interval, 1, 2, 3, 4, 5, 6, 7};
        CAN.sendMsgBuf(0x00, 0, 8, msg);
        ts2 = micros(); //interval stop
    interval=(ts2-ts1);
    Serial.println(interval);
 //    delay(10);
}
```

Listing 3. SPI/CAN send speed test code

7 Future Improvements

Further development of this device could include reducing the size of PCB by placing components more efficiently on both sides. The required I/O features can also be modified depending on requirements as also voltage dividers could be added to analog inputs to allow wider variety of sensors to be connected. Appropriate connector and enclosure are essential for usability and reliability, so those need to be considered when the design is finalized. Poor SPI performance of the microcontroller should also be considered and the options are either to make fully custom configuration files for the board. This would take some effort and it ruins the whole ease of use concept, as you need a modified Arduino IDE installation and still does not guarantee proper performance or to change the microcontroller, so better performance could be achieved with default configuration files.

8 Conclusions

The goal of this project was to prototype a CAN-node solution for data acquisition, peripherals management and control systems in race car use. For succeeding in this, certain objectives were set for the prototype. These included low cost, ease of use, light weight, small size and high performance. Most of the tasks got fulfilled but final testing revealed some issues with the performance of product. These were related to the implementation under Arduino IDE and compromised either data transfer speed or ease of use of the product, giving up either of these features is not a viable option, as these are extremely important considering the future application. This was a major disappointment, as the board was performing otherwise well.

Overall this device was successful as a first prototype despite the performance issues. All the research made for this project and most of the design solutions and guidelines are easily transferrable to another microcontroller, if in future the decision is made to change it.

References

1 Epec 38 product brochure [ONLINE]
Available at:
http://fscinc.com/documents/epec/Epec%20CANopen%20slave%20products.pdf
Accessed November 20, 2017

2 Motec SVIM module product page [ONLINE]
Available at:
http://www.motec.com/svim/ex-svim-ov/
Accessed November 20, 2017

3 McLaren SN-320 data acquisition unit product page [ONLINE]
Available at:
http://www.mclaren.com/appliedtechnologies/products/item/sensor-node-sn-320/
Accessed November 20, 2017

4 SAE homepage, Formula SAE history [ONLINE]
Available at:
http://students.sae.org/cds/formulaseries/history/
Accessed November 23, 2017

5 Metropolia Motorsport homepage [ONLINE]
Available at:
http://metropolia-motorsport.fi/
Accessed November 23, 2017

6 IMECHE, The formula student challenge [ONLINE]
Available at:
https://www.imeche.org/events/formula-student/about-formula-student/the-challenge
Accessed November 23, 2017

7 Microchip ATSAMD21G18A datasheet. 2017. [ONLINE]
Available at:
http://ww1.microchip.com/downloads/en/DeviceDoc/40001882A.pdf
Accessed November 27, 2017

8 Arduino M0 product page [ONLINE]
Available at:
https://store.arduino.cc/arduino-m0-pro
Accessed November 27, 2017

9 Arduino MATLAB support package page [ONLINE]
Available at:
https://se.mathworks.com/hardware-support/arduino-matlab.html
Accessed November 27, 2017

10 Microchip MCP2515 digikey product page [ONLINE]
Available at:
https://www.digikey.fi/product-detail/en/microchip-technology/MCP2515T-I-SO/MCP2515T-I-SOCT-ND/4307902
Accessed November 27, 2017

11 Texas Instruments TCAN332 digikey product page [ONLINE]
Available at:
https://www.digikey.com/product-detail/en/texas-instruments/TCAN332DR/296-43711-1-ND/5823573
Accessed November 27, 2017

12 ON Semiconductor NCV8402D digikey product page [ONLINE]
Available at:
https://www.digikey.com/product-detail/en/on-semiconductor/NCV8402ADDR2G/NCV8402ADDR2GOSCT-ND/3462421
Accessed November 27, 2017

13 XP Power TR05S3V3 digikey product page [ONLINE]
Available at:
https://www.digikey.com/product-detail/en/xp-power/TR05S3V3/1470-2257-5-ND/4488636
Accessed November 27, 2017

14 STMicroelectronics ESDCAN24-2BLY digikey product page [ONLINE]
Available at:
https://www.digikey.com/product-detail/en/stmicroelectronics/ESDCAN24-2BLY/497-13262-1-ND/3661760
Accessed November 29, 2017

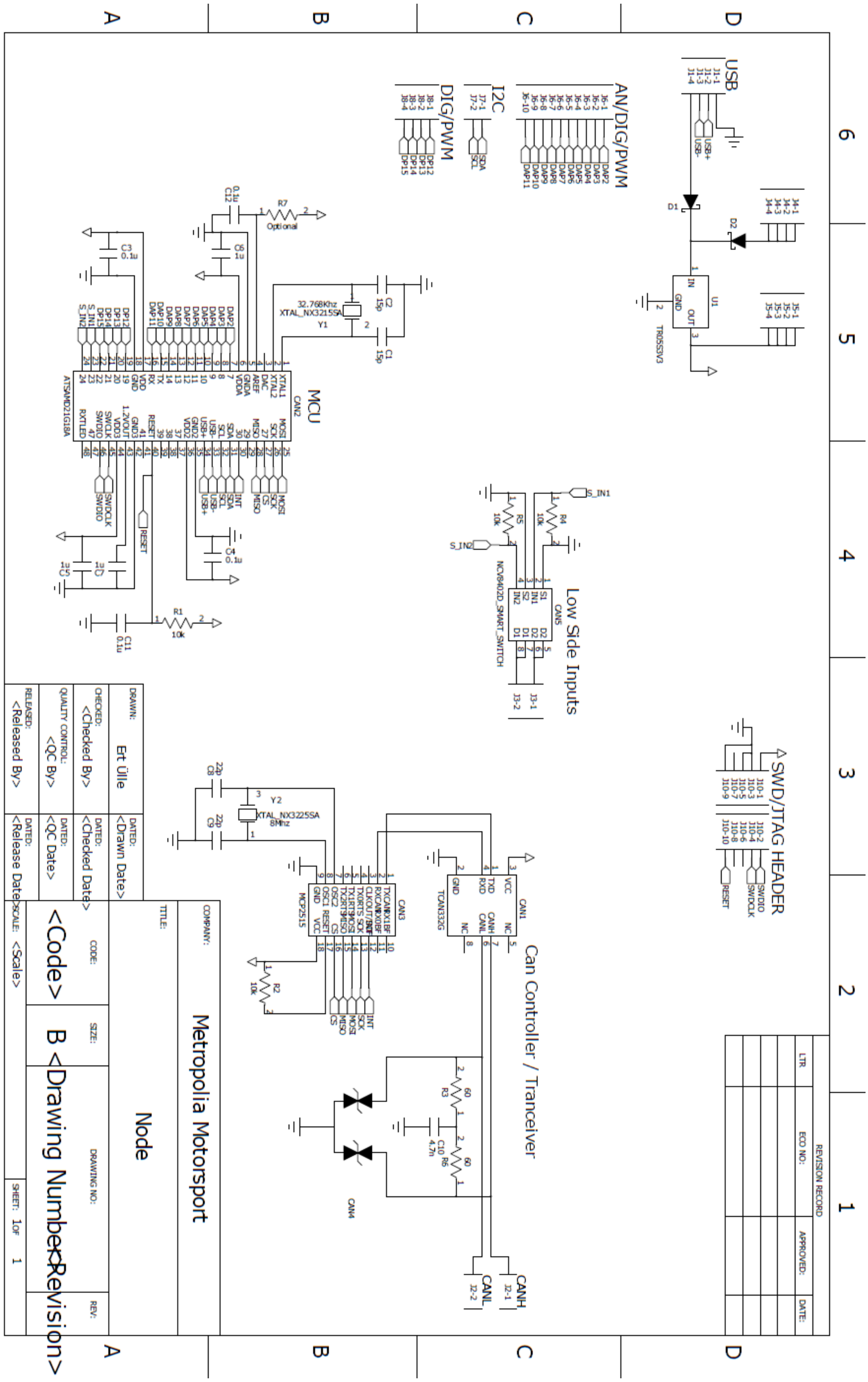
15 Microchip, Crystal Oscillator Basics and Crystal Selection. 2002. [ONLINE]
Available at:
http://ww1.microchip.com/downloads/en/AppNotes/00826a.pdf
Accessed November 29, 2017

16 Seeed Fusion PCB prototyping service homepage [ONLINE]
Available at:
https://www.seeedstudio.com/fusion_pcb.html
Accessed November 29, 2017

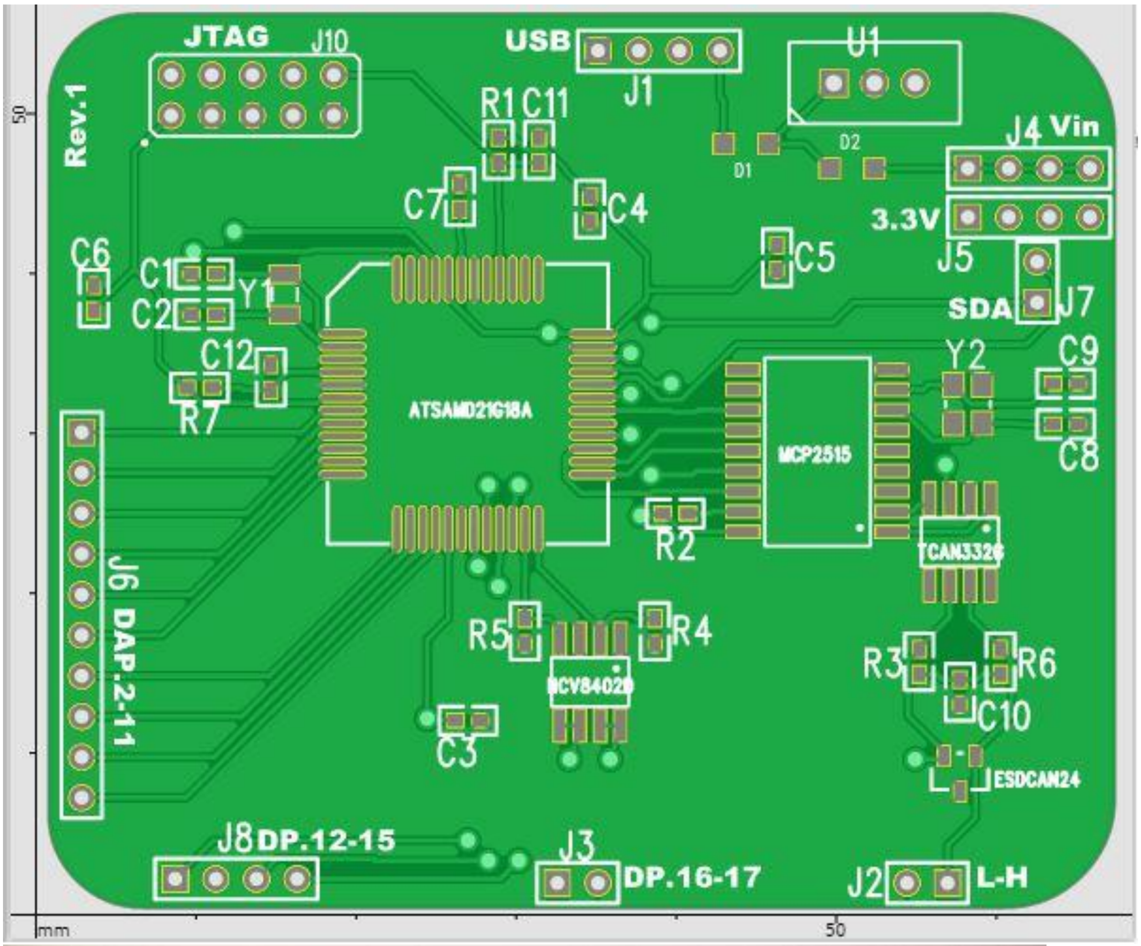
17 Texas Instruments, Introduction to the Controller Area Network. 2002. [ONLINE]
Available at:
http://www.ti.com/lit/an/sloa101b/sloa101b.pdf
Accessed November 29, 2017

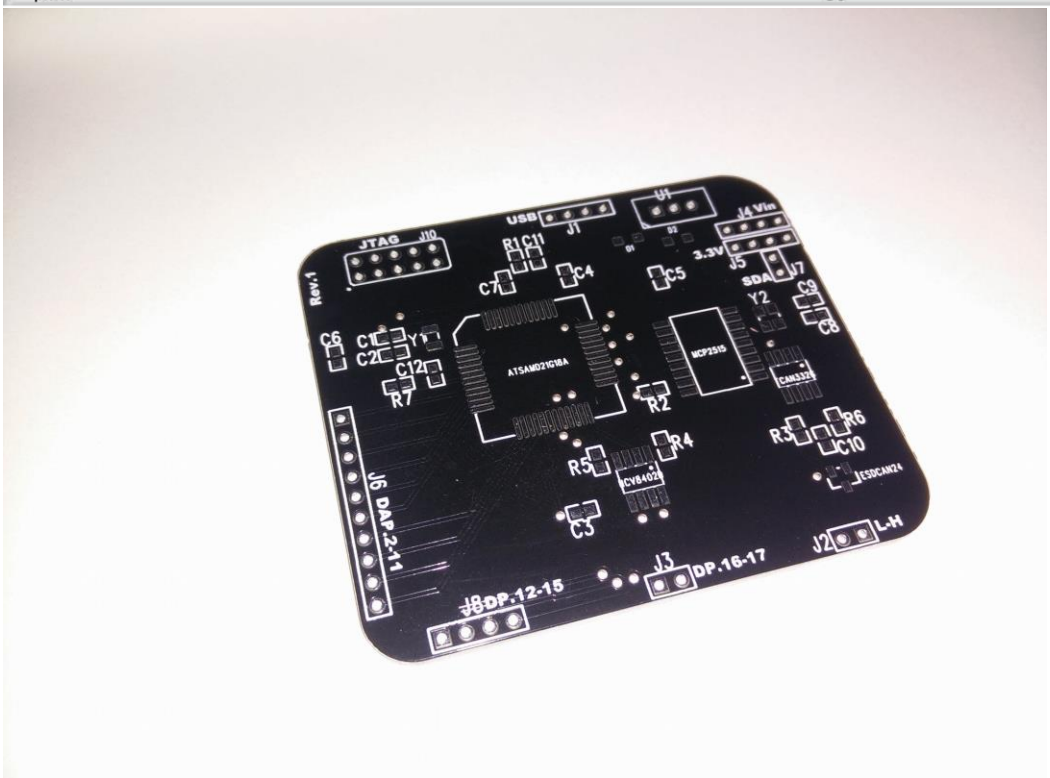
18 Wikipedia, Software and programming tools [ONLINE]
Available at:
https://en.wikipedia.org/wiki/Arduino
Accessed November 29, 2017

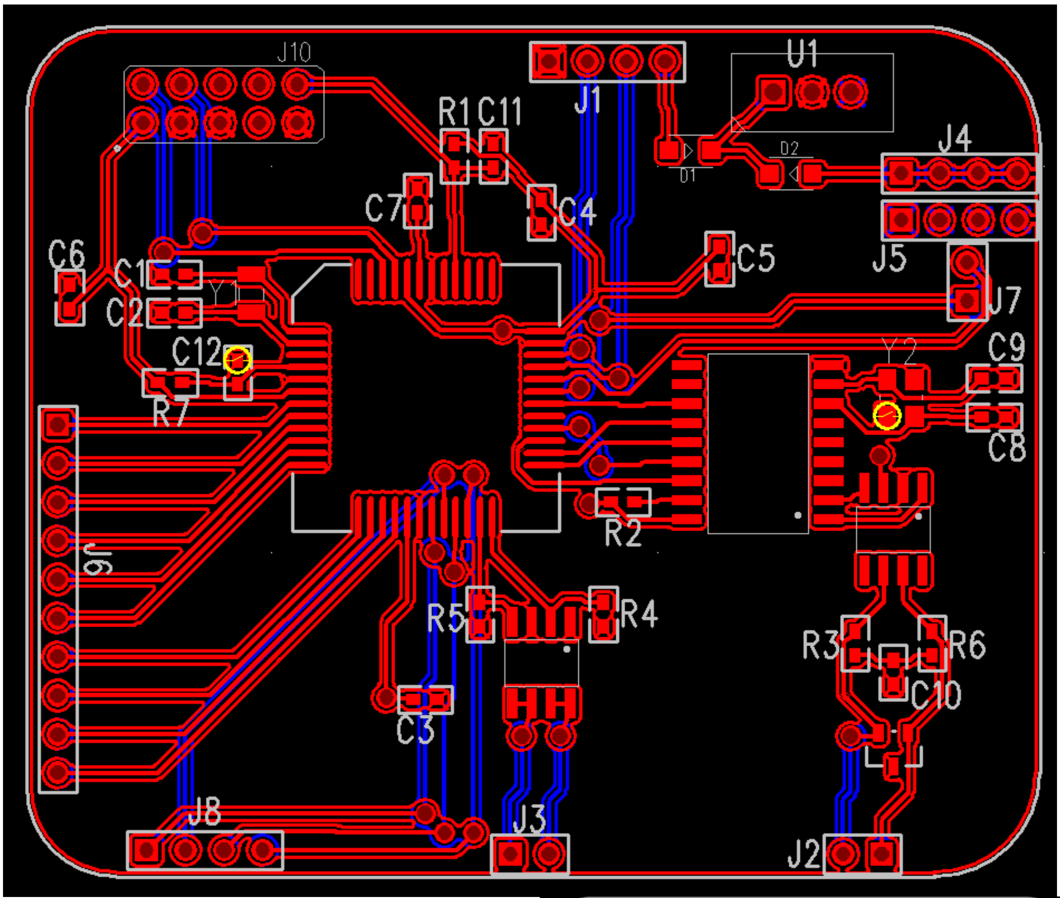
Node Schematic



Node PCB


JTAG
J10
USB
J1
U1
D1
D2
R1
C11
J4
Vin
3.3V
J5
SDA
J7
Rev.1
C7
C4
C5
C6
C1
C2
Y1
C12
R7
ATSAMD21G18A
MCP2515
Y2
C9
C8
R2
TCAN332G
J6
DAP.2-11
R5
R4
NCV8402D
R3
R6
C10
C3
ESDCAN24
J8
DP.12-15
J3
DP.16-17
J2
L-H
mm
50




J10
J1
U1
R1
C11
D1
D2
J4
C7
C4
C5
J5
C6
C1
C2
Y1
J7
C12
Y2
C9
R7
C8
J6
R2
R5
R4
R3
R6
C10
C3
J8
J3
J2