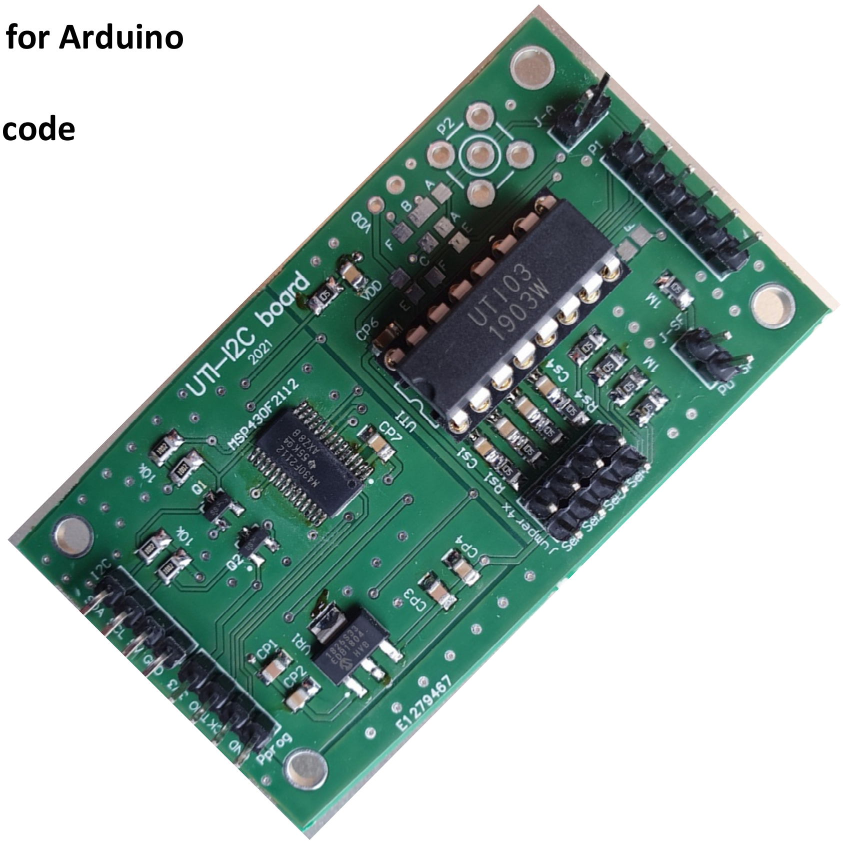
UTI - I^{2}C evaluation board for UTI



1 Introduction

The I^2C bus has existed for many years, but it was only after the original copyright expired that it started gaining momentum. Following this trend, Smartec has developed a number of I^2C compatible products. The UTI - I2C evaluation board is one of these products. Using this board, accurate sensor measurements can be easily integrated in any I^2C based system. The UTI is a highly accurate, universal, self-calibrating data acquisition system, bypassing many of the problems that normally are part of these systems and has many different application possibilities. For more information about the UTI please refer to its spec sheet. The rest of this document is about the evaluation board itself.

2 General

The output of the UTI is a multiphase pulse duration modulated signal that contains all the information about the sensor(s) being measured. These times have to be measured accurately and on this board that task is being carried out by a MSP430 microcontroller. How this is done by the processor program is not interesting for the user of this board. What is interesting however, is the fact that the numerical data available through the I^2C bus actually do represent the duration of the UTI output phases mentioned above. The not very easy task of programming the microcontroller is done by us, but the user will still have to get a thorough understanding of the philosophy behind the UTI, how exactly it works and what the data means. We again refer to the datasheet.

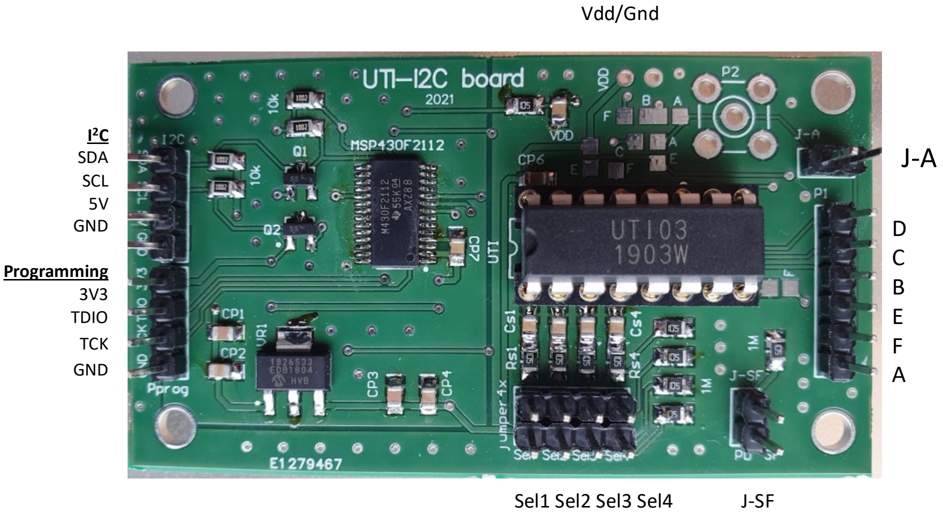


Fig. 1, The MSP based UTI - I^2C board and jumpers.

In most cases we recommend that the user first focuses on the UTI - sensor aspect. A good way to do that is by starting with our UTI usb development board, which directly connects to a PC or laptop and for which free monitoring / logging software is available. Once this is mastered, the user can then concentrate on the I^{2}C aspects, using UTI - I^{2}C board.

3 Headers and jumpers information

On the PCB a couple of headers can be found for connection purposes, as well as several jumpers, used for configuring the board. Below we explain their use.

J-A
Open: UTI point A is only connected to pin A of header P1.
Closed: UTI point A is also connected to the centre pad of P2. This allows for easier cabling when good shielding is critical, for example when measuring small capacitor values.

Sel 1 … Sel 4
The setting of these four jumpers determines the UTI mode. The UTI data sheet gives the details of each of the 16 possible modes. The presence, or not, of these jumpers present the corresponding four bits.
Open: bit is Zero
Closed: bit is One

J – SF
Open: The UTI operates in SLOW mode
Closed: The UTI operates in FAST mode, that is 8 times faster than in SLOW mode, but as a consequence the resulting measurements are 3 bit less accurate.

Vdd/Gnd
On the top of the board you find a soldering pad indicated VDD. A header pin can be soldered to this pad if desired. The pad next to it gives access to GND.

I^{2}C header.
On this header the four standard I^{2}C pins are available for connecting to the bus. The default I^{2}C address of the UTI - I^{2}C board is **0x60**.

Programming header
This header is only used for (re)programming the MSP430 which is not supposed to be done by the user.

P1 (Sensor input header)
The pins labelled A,B,C,D,E and F are connected to the corresponding UTI pins. The sensor elements, as well as additional components such as a reference element, are all connected to the UTI via this header.

Sensor input header.
The sensor can be connected to the header named A..F. for the definition of the header please consult the datasheet of the UTI. There are also some solder pads on the board for soldering references, etc, etc.

4 Hardware Schematic.

The schematic circuit of the UTI - I^{2}C board is shown in Fig.2.

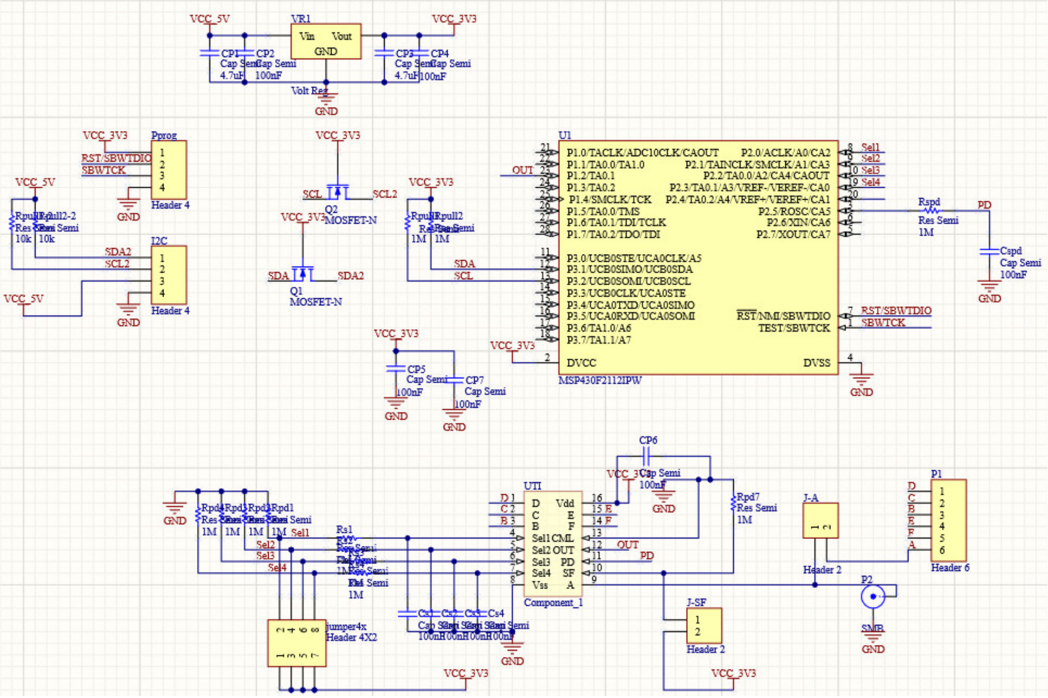


Fig.2, Schematic of the board

5 Software.

The microcontroller is continuously measuring the UTI output signal. As soon as a measuring cycle is complete, the measured times are written into a shift register. These values are then copied into the I^{2}C output buffer, unless the I^{2}C bus happens to be active. In that case the processor continues measuring the UTI phases and writing them into the shift register. As soon as the I^{2}C bus becomes inactive again, after detecting a STOP event, the I^{2}C output buffer is updated with the latest values. This way, any possible ambiguity of the output signal is avoided.

6 Reading and writing the I^{2}C device.

All communication can be done using the default address which is **0x60**. If an alternative address is needed, this can be attributed to the board by means of the command **0x10**, followed by the new address. See below. The I^{2}C output buffer is 21 bytes long. These bytes are supposed to be read in one block read action, in order to avoid inconsistencies in a multi master I^{2}C situation.

The data that will be returned is:

Function	**Toff1**			**Toff2**			**phase 1**			**phase 2**			**phase 3**			**phase 4**			**Setting *)**	**sample counter*)**	
	low	mid	high	low	mid	high	low	mid	high	low	mid	high	Low	mid	high	low	mid	high	switches	low	high
Byte	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

*)Switch settings (lower 4 bits 0000 correspond to Sel4…Sel1)
*) Sample counter low/high byte

Special commands

0x30 (no parameters)Powers down the CPU, only the I^{2}C hardware will be active after this command has been executed. The device will accept commands and return data, but will not provide new samples.

0x40 (no parameters)Powers up the CPU. The first sample after power-on may be inaccurate.

0x10 followed by the new I^{2}C device address. The new device address is active immediately after this command is executed. All subsequent I^{2}C communications should use this new device address. The new device address is not stored in non-volatile memory. A reset will revert the device to the default device address **0x60,** but it can be made permanent by one of the following commands.

0x20 or 0xA5 or 0x5A The current device address is stored in non-volatile memory.

7 Software for Arduino

For evaluation an Arduino can be used to read data from a UTI - I^{2}C evaluation board and convert it to HEX or DEC format.
In a separate file in the Engineering Notes of our website a little program for Arduino can be found. With this program the board can be connected to a PC via a Little Arduino processor board (type Arduino Nano Everywhere). The USB output of this Arduino board can be read in via a terminal program or also via the logging program for the classical USB UTI interface board. In this way a maximum flexibility is arranged.
Running a terminal program is easy and convenient. In case the UTI data has to be stored for further processing it is advised to use the logging program. The selection of the sensor is made in the I^{2}C board via straps and using the logging program this selection is set via software. **Be aware the same setting is selected in both the logging program as well on the I^{2}C interface board.**

8 Ordering code:

UTItoolkitI^{2}C	Toolkit with I^{2}C output
UTI DIL16/SOIC18	Universal Transducer interface

In the engineering notes UTI part:

Software for read and store capacitance and resistive output