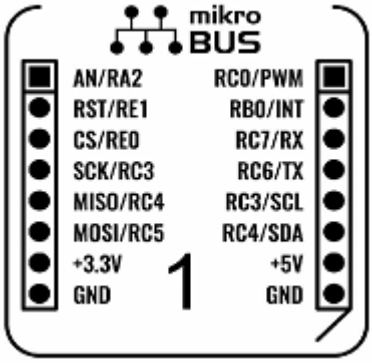
Click identification functionality

mikroBUS standard

What is mikroBUS

The mikroBUS standard defines mainboard sockets and add-on boards (Click) used for interfacing microcontrollers or microprocessors (mainboards) with integrated circuits and modules (add-on boards).

The mikroBUS socket comprises a pair of 1x8 female headers with a proprietary pin configuration and silkscreen markings. The pinout (always laid out in the same order) consists of three groups of communications pins (SPI, UART and I2C), six additional pins (PWM, Interrupt, Analog input, Reset and Chip select), and two power groups (+3.3V and GND on the left, and 5V and GND on the right 1x8 header).



Detailed description of mikroBus standard

What is Click

Click is add-on board compatible with mikroBUS socket.

How it works

On main board with host MCU can be one or more mikroBus sockets. Some MCU host pins can be shared between more mikroBus sockets(Ex. SPI, UART, I2C). Host MCU communicates with **Click** on mikroBus socket through available pins.

Click identification functionality

What we want

Current available **Click** boards don't have any possibility of identification, so host MCU can't detect which **Click** board is placed on mikrobBus socket. Ex. If firmware on host MCU is written to works with one **Click**, but another **Click** is placed on mikroBus socket it can lead to unpredictable behavior.

We want to add click identification functionality, so host MCU can detect which click board is placed on corresponding mikroBus socket. Also, we want to retain backward compatibility, so that all of currently available **Clicks** can work like they work now.

How we plan to implement it

On new **Click** board (**Click with ID**) we plan to add identification MCU (***ID MCU***) which will be used to store relevant information about **Click** board. Communication with ID MCU must be through pin(s) on mikroBus socket. We must have way to address all mikroBus socket and read corresponding Click ID information.

How it works

Clicks with ID after power-up works as old **Click** board without ID MCU. (**REGULAR MODE**). On this way **Click with ID** will work with firmware which was written for old click board.

Host MCU with firmware which has capability to read ID, first must put **Click with ID** in **ID MODE** state in which is possible to read ID information.

To put **Click with ID** to **ID MODE** host must send pattern through MOSI line while SCK line mustn't change state during this interval. On this way, we secured that no-one regular SPI communication accidentally place Click to **ID MODE**.

Current pattern on MOSI line to put Click to **ID MODE** is four clock cycle with period of 1KHz. This pattern is used for testing and can be changed before product release.

SPI lines can be shared between more mikroBus socket, so all clicks on mikroBus sockets which share this line will enter in **ID MODE**

To address mikroBus socket and read corresponding ID we use CS line on mikroBus socket, because that is only one host line which is unique for all ours mainboards per mikroBus. On this way, host can address all mikroBus socket on the mainboard and read corresponding ID.

Protocol which is use to read ID from Click board through CS line is standard **1-wire**.

Why do we need two modes

It looks that we can read ID only using CS line while SPI bus is inactive. Example: If we communicate through CS using 1-wire protocol while SCK line doesn't change state we don't activate SPI communication with SPI slave on **Click** board. But on some already produced **Click** CS line is not only use for SPI communication. This line can be use for various needs. Example as GPIO line for motor control. So reading ID can activate some functionality on slave. This is allowed by mikroBus standard. That is reason because we must first put **Click** to ID Mode to unbind CS line from slave on **Click** board.

Revision Number	Revision Date	Summary of changes	Authors
0.01	January, 2021	Initial version	N.M