

MAX31343SHLD RoadTest | Review of MAX32625 MCU and MAX32625PICO board



[misaz](#) 16 Jul 2021

I welcome you to my review of MAX31343 Shield. My review is split to multiple chapters linked in following table of contents. Main page of review (marked with yellow star) contains summary, final score, and description of all chapters. Most important chapters of review begin with “Review” in name and contains detailed description of thing under review. This chapter is one of them. Except this chapter reviewing bundled PICO board and MCU present on PICO board, there are also chapters reviewing RTC chip and evaluation shield. There are also described projects which I have done for evaluating this shield and finally there one tutorial showing how to easily use roadtested shield with Arduino.

Review Table of Contents

- [Introduction, Overview of Table of Contents, Summary and Score](#)
- [Review of RTC Chip](#)
- [Review of Evaluation Shield](#)
- Review of MAX32625 MCU and PICO Board (this article)
- [Tutorial: Using Shield with Arduino](#)
- [Project 1: Platform Independent Library](#)
- Project 2: Time Syncing over BLE
- Project 3: Linux Driver

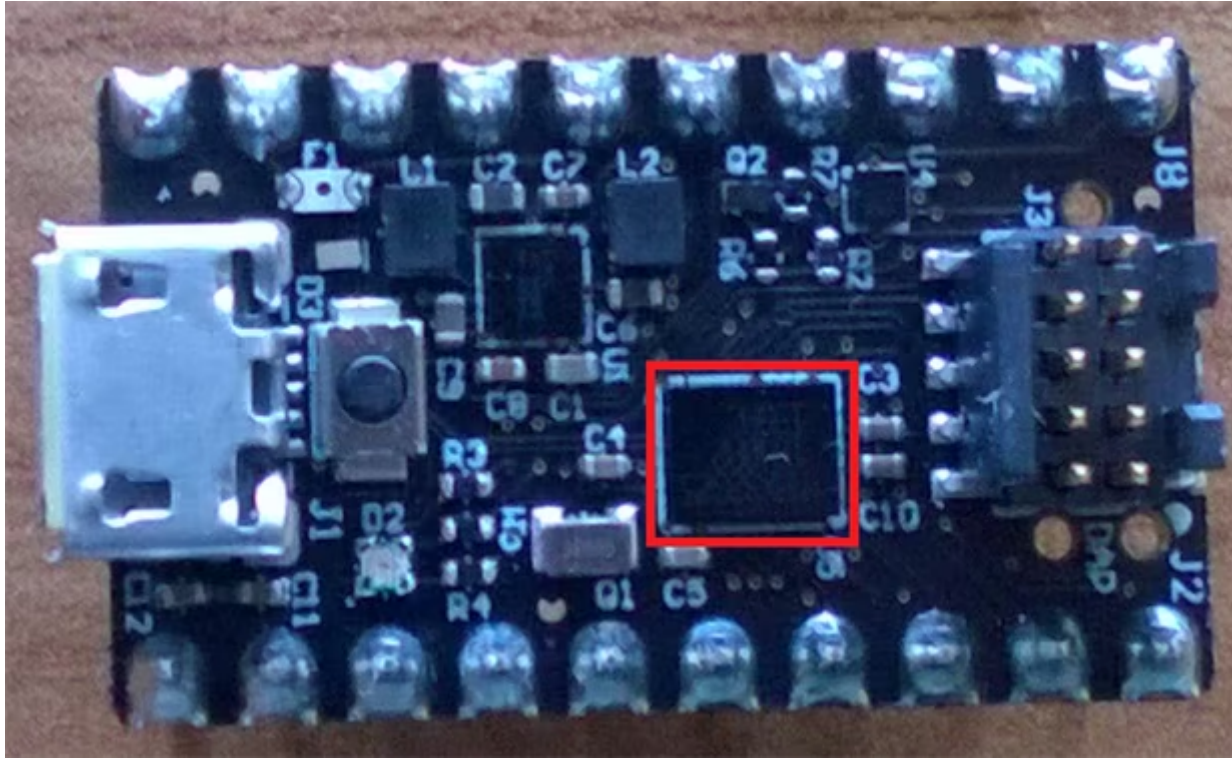
Review of MAX32625 Microcontroller and MAX32625PICO board

As part of review I decided to briefly review bundled PICO board and it's MCU also. Bundled PICO board is not designed to running customer program, but it is flashed with software allowing you to evaluate MAX31343 chip on shield over USB connection by provided GUI tool. While it is not very easy as described later you of course can connect SWD/JTAG adapter to this board and program it with your own firmware. At beginning is good to say that PICO board is not designed to doing this. PICO board is designed mainly as very small universal board which is used for evaluating Maxim's other products and can be used as DAP link for programming and debugging other board. If you want to run your code on this board, you can deploy it using bootloader or over SWD/JTAG connector which is not soldered by default. Bootloader approach simple but requires pressing button and repowering chip every time when you want to deploy new code and do not support any kind of debugging. In opposition SWD is fast and allow debugging but it is hard to access SWD/JTAG port for programming MCU on this board. Note that 10-pin Arm Cortex connector is used to program external board and not an onboard chip. It is quite pitying that board is designed in way that is very hard to program it because it features powerful and very flexible MCU and peripherals. Board with these features which I describe later efficiently beats many other bords at extremely less space occupied. At first, I will describe chips on PICO board, then I will briefly describe it, compare it to other comparable small boards and later in this chapter I will describe it's MCU.

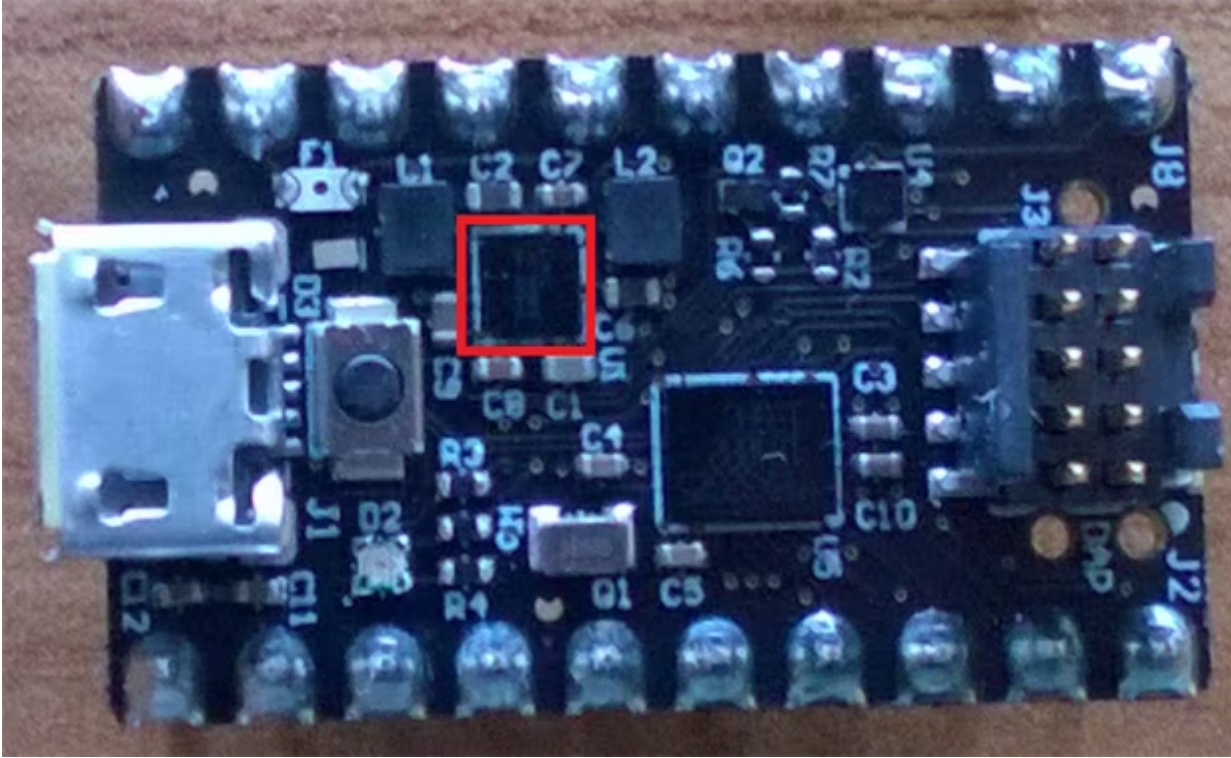
MAX32625PICO Structure

While board is very small and board is one of the smallest boards with MCU at market, it features 3 chips, lot of passive components, USB connector, 10-pin SWD connector for programming external MCU when used as DAP link, crystal and RGB LED. These are lot of components when you take in account size of board. Very interesting are 3 chips.

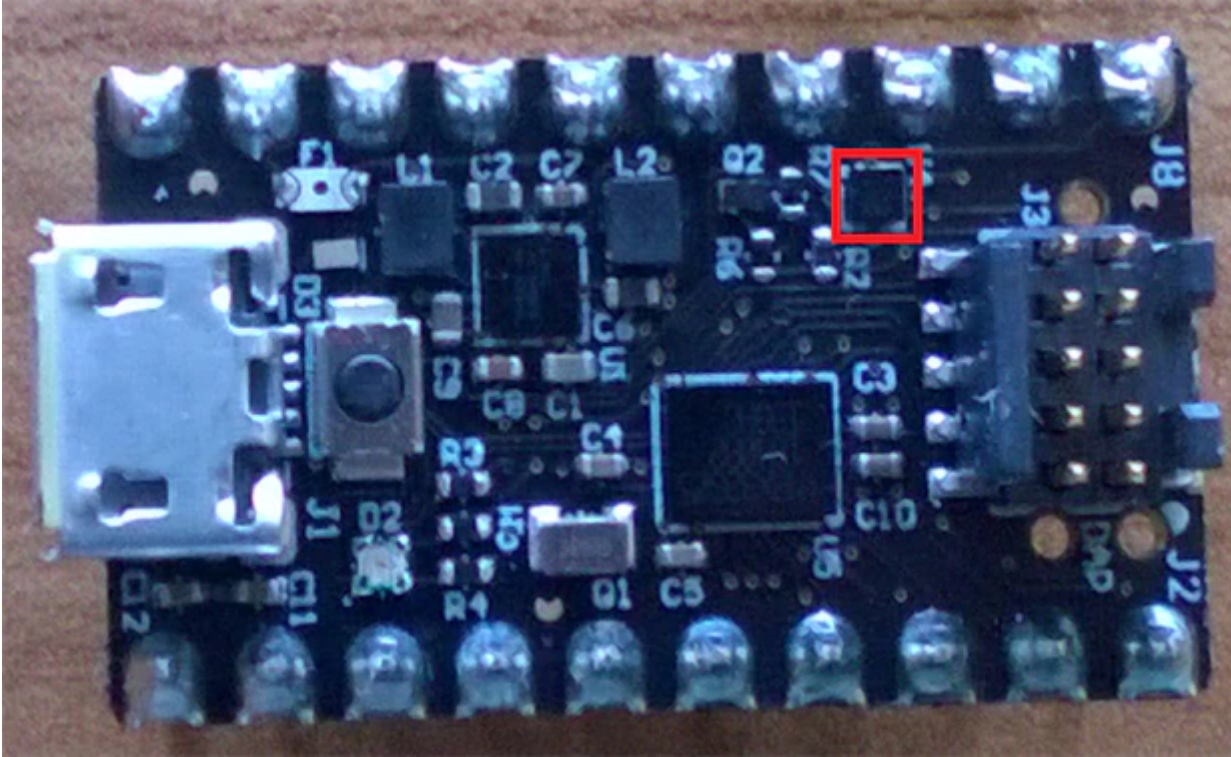
The largest chip is MAX32625 microcontroller. It looks large only on image. In fact, it is $3,8 \times 3,0$ mm and has 63 pins at the bottom of chip! MCU is highlighted on following image:



Second largest chip is MAX14750 and it is chip that integrated multiple regulators converting 5V input from USB to three voltages - 3.3V, 1.8V and 1.2V. It has many passives near it because some their regulator requires them. But when considered that it creates 3 different voltage levels, PCB space occupation is even when included passives very small. PMIC is highlighted on following image:



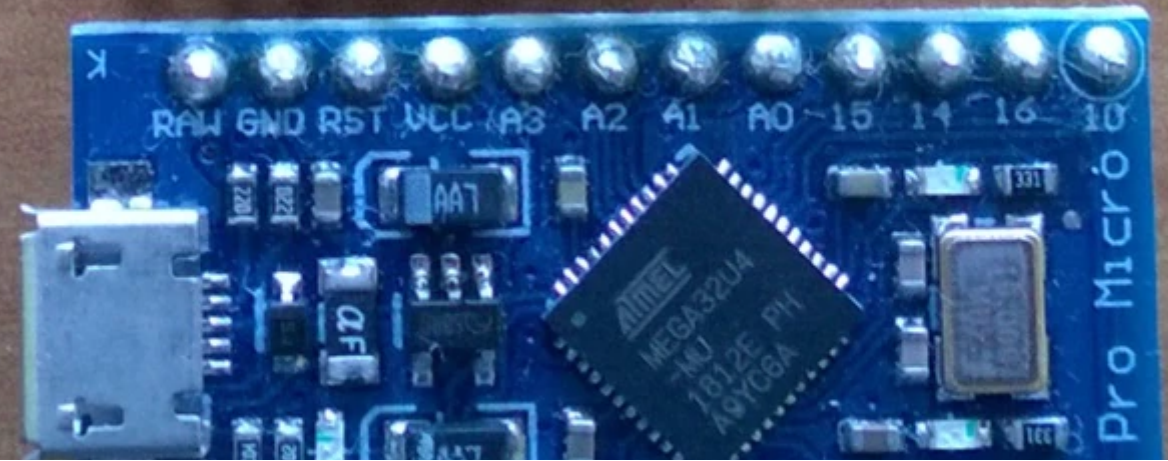
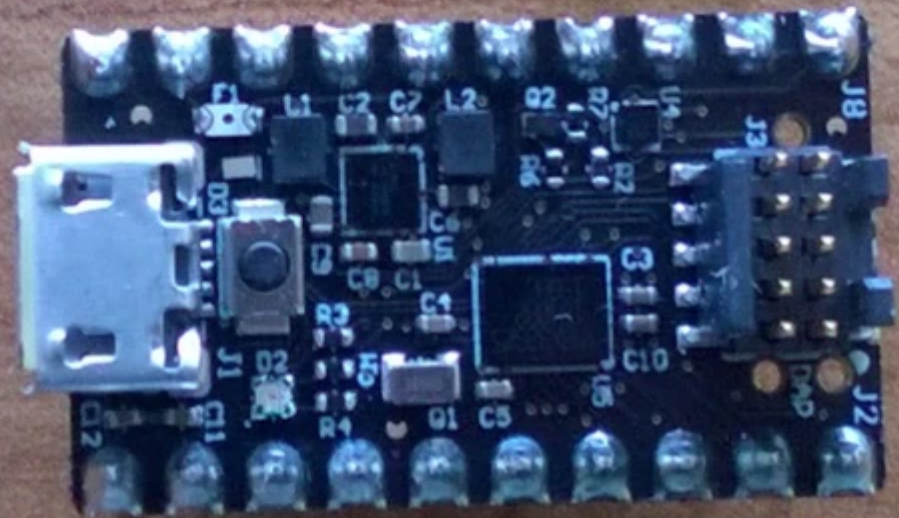
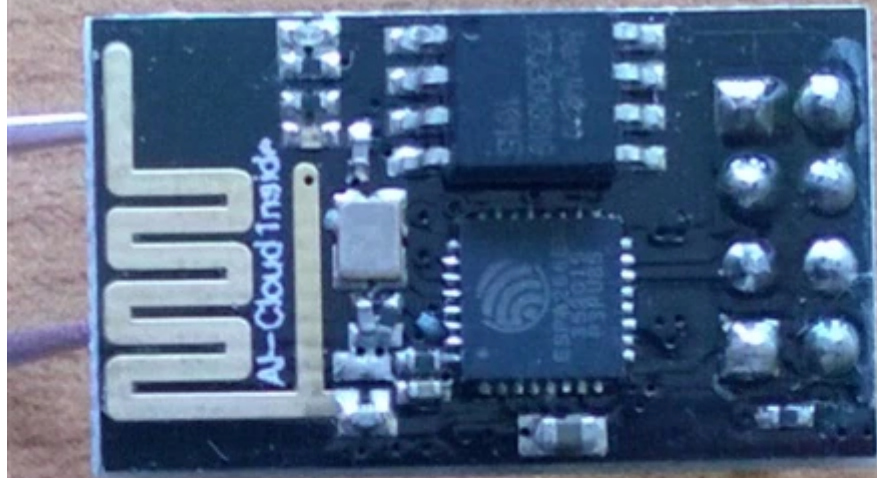
Last chip (the smallest) chip is MAX14689 which is analog switch and it is used for modification pinout behaviour on two pins. This is quite simplified description of its purpose, but fully complete description would be too long. This chip makes board even more flexible than it is. It is highlighted on following image:



MAX32625PICO Comparison

Board is very small. On the following photo it is placed next to ESP8266 on ESP-01 module and clone of Sparkfun Pro Micro. It is very similarly sized to ESP8266 module but has much more pins exposed, also has exposed USB connector (which is not supported by ESP8266) and even on the same size contains regulator which can provide more flexible powering options for external components. But this comparison is not very clear because it compares totally different products targeting totally different applications. Much better is comparison with larger Sparkfun pro Micro. This board also supports USB, has lot of GPIOs but has much slower MCU with less peripherals, is less energy efficient and without modification or external level convertor supports only 5V components. In comparison with these boards MAX32625PICO looks like very flexible and powerful board and this I agree with it. The only disadvantage is that is quite complicated to program it and develop firmware for this MCU. This is most probably strong benefits of other two mentioned because they are programable simply using Arduino. In opposition MAX32625PCIO is only board

with SWD/JTAG support for debugging which requires additional skills and need for owning some debug probe. Also note price of board. It is 13.20 USD at the time of writing at Newark while price of less powerful and flexible Pro Micro is 17.95 USD at sparkfun website.



MAX14750 Power management IC

PICO board features power management IC (from Maxim) which is designed to power MCU. MCU has quite complicated requirements and this chip satisfies them. MCU must be powered by at least two voltages. One is 1.2V and this power is used for powering chip core and second power supply must be 1.8V for powering some other parts of chip and GPIO ports. Lastly some pins of chip can be powered by voltage higher than 1.8V and in this case this voltage must be bring to chip also. PMIC supports 3 regulators. I original thought that all of them are configurable using I2C but this there are used variant which cannot be reconfigured at runtime when powered on.

MAX32625PICO Schematics and board flexibility

While board is extremely flexible it may be considered as a disadvantage also. It is very hard to understand all concepts used on this board. Schematic which is present in datasheet is also not a very clear. Chips are separated there to multiple components. Wires are named but they are still routed over whole schematic paper. And finally, it is not easy to understand some concepts like purpose of connection analog switch at the first time. It took me about hour to understand all concepts used on this board. I also do not like when schematics are part of datasheet because I like having opened datasheet and schematic at the same in two instances of PDF viewer. Schematics in datasheet is also 90deg rotated which make even more complicated searching in datasheet and looking to schematic at the same time.

MAX32625PICO Documentation

MAX32625PICO documentation is good. In comparison to documentation for MAX31343 chip and shield it looks consistent. Datasheet is nice looking and describe almost all features of board. It has also illustrative pinout drawing which also contains pins used to drive onboard peripherals like PMIC and analog switch for modifying pinout behaviour. MAX14750 PMIC has also good documentation but in comparison with MAX31343 it has not described access level of registry fields and has not generated navigation for PDF reader. I looked it only briefly, but it looks complete and detailed enough.

In further part of this article I will briefly describe MCU presented on board.

Classification of MAX32625 MCU

At beginning of MCU review I usually describe which chip I describe and it is assigned to MCU families from vendor but in this case it is complicated because Maxim do not use hierarchical naming system or I do not understand it. Only note which I can say is that MAX32625 and MAX3262**6** are siblings and difference between them is only about that cryptography accelerators, secure key storage and some other related security features are present only in variant ending with 6 in name. Otherwise the chips are the same. There is also datasheet for MAX3262**0** and MAX3262**1**. They are like MAX32625. While they are numbered lower, they have more memory and peripherals otherwise diagram looks the same.

Maxim offers more differently numbered MCUs. They also provide MCUs with Bluetooth Low Energy support and provides MCUs with some AI accelerators. When I was searching for Maxim MCU, I found MCUs with Arm Cores, Maxim proprietary cores and MCUs with RISC-V coprocessor. Many of them looks very good but it is hard to find and compare them due to lack of any selector, helper, or summary table on their website.

It is hard to estimate but it looks that is MAX32625 is quite old in their offer and newer MCUs like MAX326**5x**, MAX326**6x**, MAX326**7x** and MAX78000 usually offers more features. They have significantly more readable documentation mentioning newer date of first revision of document and some of them have also reimplemented some peripherals. Some concepts remain same over

all Maxim MCUs but many peripherals which I have seen in previously datasheet of chips from mentioned families significantly differs from roadtested MAX32625.

Clock subsystem

MAX3265 has fixed clock subsystem consisting of internal 96 Mhz oscillator generating clock for CPU core and I did not find any way to connect external high-speed oscillator. There is only way to connect 32 kHz low-frequency oscillator which is used for clocking building RTC, watchdogs and some other low-speed peripherals. There are very few options how to configure clock subsystem. Benefit of this simple design is that you do not need to worry about clocks because there are only 96 MHz clock which can be simply prescaled and this clock is used in almost whole system. Disadvantage is that you cannot reduce clocks in a flexible way which for example Cypress PSoC offers and possibly reduce power consumption by reducing clocks in more efficient way. But for most application this design is sufficient, so you do not need to worry about it much. There is additional 4 MHz clock which can be used to reduce clock from 96 MHz to 4 MHz for lowering power consumption. The good feature is that 96 MHz oscillator can be calibrated using low frequency 32kHz oscillator, so while it is inaccurate by default (2% accuracy) it still can be used for example for USB operation.

Performance

CPU Core is Arm Cortex-M4 with floating point unit. As described in previous section It is clocked at 96 MHz. This should be enough for most applications. There are also DMA which can significantly improve performance. It is quite complex DMA driven by commands stored in memory. Except copy operations it can do much more. You can implement complex operations using this like jumps and branches which efficiently allows you write some programs for a DMA engine. You can also execute multiple of these programs at parallel. Chip supports 6 of these execution engines.

Peripherals overview

MAX32625 offers 6 timers and many common peripherals like 3x UART Controller, 3x I2C Controller, 4x SPI Master controller. There are also SPI controller designed for connecting external flash memory. This controller supports mapping content of attached memory to address space and directly execute code from this memory. MAX32625 supports 10-bit ADC. It also has AES encryption accelerator and CRC calculation unit. There are also USB controller which supports device mode. MCU has integrated RTC block and have 3 watchdogs of two types. Later in this section I will describe some unique peripherals (which I did not mention yet) and I will point some peripherals which I miss but can be found in competitors MCUs.

All peripherals are designed in very complex way. As a result of this they are usually very easy to use. I have seen many peripherals designed in way that you send some encoded command to it and that is all. All other stuff is handled by peripheral in background. This is fine because you can implement it very easily, but there is some amount of mystery what is happening in the background and also possible limitations. But I never run into any issues with this when trying features of this MCU.

Unique Peripherals

There are two peripherals and one feature which I consider unique. I will describe them in following sections.

1-Wire bus driver

Because Maxim acquired Dallas Semiconductor, they now have access to large portfolio of 1-Wire bus devices. 1-Wire bus is interesting bus oriented to reducing number of required wires needed for connecting slaves to the bus. It reduced bus lines to single wire and even more, it is designed in a way allowing parasite power supply over this single data wire. For connecting sensor, you efficiently need only two wires at all – GND and one data wire. You do not need Vcc line. In comparison with other buses, specification define quite complex system of device identification (globally unique identifiers are used) and bus enumeration process

which makes this bus even more interesting. Disadvantage is that it is quite proprietary bus and very few vendors supports it in comparison with for example I2C bus.

Controlling 1-Wire sensors using microcontroller is usually handled in software. Because bus is timing sensitive (because it has no clock line) this is implemented using disabling interrupts and short delay usually implemented using busy waiting loop. It works but it is not ideal. MAX32625 has HW accelerated peripheral for controlling this bus. I have not seen 1-Wire driver peripheral in any other MCU from different vendor. Most probably it is cause by proprietary nature of the bus. Peripheral works in a way the you issue commands what you want to transmit on the bus and peripheral do it in background. You can run your own code at the time when transaction is progress and even improve overall system performance. I tried this driver and it works very well. SDK contains driver for it. You can drive it at low level (sending bits, generating reset sequence and so on) and at higher level (generating transaction according to defined protocol – reading scratchpad, writing it, and so on). Bus enumeration is implemented in SDK, so it is very easy to use.

Pulse train engine

Second interesting peripheral which is rarely seen on other microcontroller is pulse train engine (PTE). Pulse train engines are used for generating pattern on output pin. You can use it for similar purpose like you can use PWM, but you can generate more advanced patterns. It can be used for smooth and precise motor driving which is most probably one of main use cases of this feature. Peripheral support flexible timing configuration. For an experiment I tried to implement generating valid UART signal using this peripheral and it really worked. MAX32625 supports 16 of these engines and you can connect output of them to one or more pins but there are restrictions what can be attached to which pin. It is interesting feature and if you use motors in your design or find other use case it may help you.

Multiple Voltage Levels on IO Pins

Last feature which I consider quite unique while it is not totally unique is that chip supports two IO powering voltage levels and even more – you can select which of them will be used on every single pin independently. This feature is usually supported by high-end MCUs with very high pin density but usually is fixed to banks. You usually can power one group of pins by one voltage (let's say Vddiobank1) and second bank of pins by different voltage (let's say Vddiobank2). Maxim do not use banks concept (at least from user perspective) and do not restrict this feature to high-end and high-IO port density chips (MAX32625 has only 40 GPIOs). MAX32625 has two IO port powering voltages (Vddio and Vddioh) and you can choose in software which one is used for powering every single pin. You for example can bring 1.8V to Vddio and 3.3V to Vddioh and then you can select in software which voltage (1.8 or 3.3V) is used for logic on for example pin P2.5. You can make your decision independently for example on configuration of sibling pin P2.4. This configuration is usually not possible even on chips which support multiple powering voltages (per bank).

If you need to mix 1.8V and 3.3V chips in your design this feature may significantly help you and save money for external level converters and so on. While chip supports any IO voltages between 1.8 and 3.3 volts there are no 5V tolerant pins which can find in competitors' chips.

Missing features

I classify MAX3265 as mid-range chip so this mean that you cannot expect some peripherals from high-end chips. MAX3265 has no camera and display interface. It has no support for advanced buses like CAN, Ethernet, or interface for SD cards. There are also no audio related peripherals like I2S and PDM controllers. Security engines and secure key storage is part of MAX32626 only.

It is quite pitying that Maxim do not integrate any advanced analog peripherals into their MCUs because they can design and offer best in class analog components with fantastic parameters. MAX32625 has only 10-bit ADC. There are no DAC which we can see in some newer MCUs from other vendors. There are no analog comparator and so on. Similarly, there are no support for capacitive touch sensing which we can see in some newer MCUs from other vendors. I can imagine many other analog peripherals which Maxim can do and possibly can be implemented as part of MCU. Maxim is very good in designing analog chips, so it is interesting to

me that they do not includes these functions to the MCUs. I think this is big opportunity to Maxim because they can design all of them and their analog chips offered on website has very good parameters.

Development Environment and SDK

Maxim provides slightly modified Eclipse. Maxim also includes some useful plugins like registry viewer showing content of peripheral's special function registers in default Eclipse instalation. It is based on SVD files which are also provided. Otherwise it is plain Eclipse. IDE is usable and works well but for example STM32CubeIDE is still integrated deeper and contains more useful tools like Hard Fault Analyzer. But I think that quality of IDE provided by Maxim is good when you take in account, that MCUs are not their core business and they produce only few MCUs in comparison with other vendors. Maxim's IDE is hard to find on their website. It is linked only on some pages and for some reason it is not linked at MAX32625 page. I found them randomly when browsing over their website.

It is hard to create (working) empty project. While there is some empty project, wizard need to specify development board used and PICO is not present there. I created my projects by copying example code. This approach works well. Project structure is simple. It is Makefile project, it is easy to understand, structure makes sense. Because of simplicity it is also easy to manually setup compiling environment and compile from command line.

As part of SDK there are libraries for driving peripherals of MAX32625 chip. Libraries are good. They properly handle errors, have lot of comments and are easy to use. There are also some doxygen generated documentation but instead it I recommend browsing over provided code samples which are also very readable and good quality. Library for driving peripherals is named MAX32625PeriphDriver and there are some additional related libraries like MAXUSB. MAX32625PeriphDriver is very similar to ST's Standard Peripheral Driver which is on ST platform still available but was suppressed by newer HAL and HAL LL libraries. You may know this style of libraries if you have worked with ST's MCU before HAL was introduced. Only note is that MAX32625PeriphDriver Library looks like it is not maintained anymore. Comment mention copyright from 2016 and it looks like that library was not updated

since 2016. This most probably also apply to already mentioned IDE which comes with the SDK. Probably this is not a significant issue because there are probably no needs from Maxim side to update code which is already working, but when working with library you are feeling that you work with something old.

Toolchain provided by Maxim uses GCC for compiling and OpenOCD for debugging. Because MAX32625PCIO has no onboard debugger, you need to use external one. Connection and basic software setup I did in the similar way as I have shown in as part of my previous RoadTest review about Arduino Nano 33 BLE Sens and especially nRF52840 chip ([Debugging nRF52840 Using Raspberry Pi](#)). I originally wanted to debug using ST-Link instead of more complicated setup using Raspberry Pi, but this was not possible due to some complication described later. Also note that SWD connection support only 1.8V voltage level so you need to use level converter between MCU and your debugging probe unless it supports 1.8V logic naturally. The problem with flashing and debugging MAX32625 is that while mainline OpenOCD looks like it supports MAX32625 chip, it does not work. I successfully backedup original firmware from PICO board for possible reverting, successfully erased MAX32625, but when I want to flash my new firmware to it, it crashed. After a while of troubleshooting I decided to try OpenOCD provided by Maxim instead of using mainline one. This raised another problem, because Maxim provide older OpenOCD and removed configuration files for debug probes which they do not support. OpenOCD provided by Maxim has configuration files for CMSIS DAP ([link](#)) and FTDI chips. So, you cannot use for example ST-Link as a debugging probe just because there is no working configuration for provided OpenOCD. Luckily, I own board with FT232 chip with required ports exposed, so I created my own configuration for OpenOCD for this chip and used this configuration for accessing SWD of MAX32625. Using this complicated setup after about 2 hours of troubleshooting and experiments I was able to flash (and debug) MAX32625 chip on PICO board by external FTDI FT232 and OpenOCD provided by Maxim. Because IDE is eclipse, I was able to easily reconfigure it and I was able to deploy new firmware and start debugging just by clicking green bug button in IDE.

MAX32625 Documentation

I have big issue with MAX32625 documentation. There is datasheet, which is good, contains information that you expect from datasheet and is easy to read. In the other hand there are document named User Guide. This document contains information which you usually find in documents that other vendors name Technical Reference Manual or just Reference Manual. This document is bad. It is very bad. It is PDF but for some reason page size is not standard A4 format, but it is some landscape size. This is not a big issue, but it is at least interesting. Document is formatted in company colours. Document contains text description of peripherals and description of registers. Register descriptions are crazy. Registers description has no overview table which significantly improve readability and which you may know from similar documents from other vendors. Instead of it they are covered by subchapter which contains fields described as another subchapter. It is very hard to distinguish what register name or field name is. Registry descriptions in this document I consider completely useless because it is very hard to find anything in this structure. Only good part of this document are text descriptions which are good, but they are not comprehensive and sometimes they also lack important information. For example, there is described that I2C master controller is driven using commands pushed to FIFO, but there is no description how commands looks like, their encoding and so on. Possible listed in provided library only.

Documentation inconsistency which I already mention in documentation section of chapter Review of RTC chip also apply to documentation of MCUs. While User Guide (Technical Reference Manual) is very bad for MAX32625 other MCUs in families MAX3265x, MAX3266x, MAX3267x looks as standard TRMs from other vendors and are useful in comparison with MAX32625 User Guide. In comparison to MAX32625 User Guide they contains more readable registers description formatted in table instead of chapters and subchapters. After some research I think that MAX32625 was one of the first MCU in their MAX326 family, but I am not sure why they did not rewrite the documentation yet.

Availability and Price

Official listed price of MAX32625 MCU at Maxim website is depending of package variant rated at 3.92 USD for 256 KB FLASH variant of chip and 4.74 USD for 512 KB FLASH variant. These are good prices for this chip. But distributors for some reason sells these chips for between 9 to 15 USD and this I consider highly overpriced. It is quite funny because some variants of chip costs

about 14 USD which is more than is price of whole PICO board (which cost is officially listed for 10 USD at Maxim website and it is about 13.5 USD at distributors). Down to prices about 6 USD prices decreases when you buy batch of about 2500 PCS.

The availability of MAX32625 and PICO board is very good similarly to MAX31343 as mentioned in chapter Review of RTC chip. Most distributors have all these chips and boards in stock even there are components supply outage at the time of writing this review. Maxim has assured supply chain very well.

Summary

MAX32625PICO board is fantastic extremely powerful and flexible board. It is the most interesting and smallest board at the same time which I have seen in this category of boards ever. MAX32625 is technically good mid-range MCU with some unique features. Biggest disadvantage of using this chip is quality of documentation which is very bad. If you want to develop using Maxim MCU, I recommend looking for newer families which has much better and more standard looking documentation and supports new features like Bluetooth Low Energy or AI accelerators.

Top Comments

3 comments 0 members are here



[dougw](#) over 2 years ago +3

It is interesting that it has 1-wire hardware. I still have a Java Ring and it is on my bucket list to see if I can still communicate with it.



[dougw](#) over 2 years ago

It is interesting that it has 1-wire hardware.

I still have a [Java Ring](#) and it is on my bucket list to see if I can still communicate with it.



[dang74](#) *3 months ago*

Amazing and very in depth review. I was impressed by the unique features you mentioned, especially the ability to select different voltage levels for individual I/O pins.



[misaz](#) *3 months ago in reply to [dang74](#)*

Thank you for feedback. 😊

Yes. It is cool especially when working with both 3.3V and 1.8V devices in system.