Tim's Blog

The “terrible” 3 cent MCU – a short survey of sub $0.10 microcontrollers.

Like many others, I was quite amazed to learn about a microcontroller sold for only 0.03 USD via the EEVblog last year. How was this possible? Many assumed this was a fire sale of an old product. Digging a bit further, it became apparent that there is an entire market segment of ultra-low-cost microcontrollers. Almost all of them are products of rather unknown companies from China or Taiwan. This write up summarizes my findings in this rather peculiar niche.

We already learned that there is a large variety of very powerful $1.00 microcontrollers, but what about the $0.10 MCU? Are they indeed all “terrible”, as suggested elsewhere?

Methodology

How to define a $0.10 microcontroller? Any way you put it; it will be a somewhat arbitrary choice. I took a straightforward approach and used the price of the 100pc bracket at LCSC. Six vendors where identified that had one or multiple devices below $0.10, all from Asia. If different packages types of the same MCU were available, I picked the SOP8 version. Some manufacturers had more than one candidate, so I had to limit myself to representative devices for a final candidate list of eight.

I was not able to find any sub $0.10 MCU at the large distributors like Digikey or Mouser. Just to state the obvious: This does not necessarily mean that it is impossible to find sub $0.10 MCUs from western manufacturers with the right order size. Two factors seem to come into play here: First, LCSC seems to operate on much smaller margins than the established distributors. Secondly, the established MCU manufacturers are not as reliant on smaller customers and can therefore command a premium on low volume orders.

Due to lack of programming tools and evaluation boards I was only able to review most devices by datasheet, with the exception of the Padauk MCUs.

Overview

In total, eight candidates from six different manufacturers where identified. A summary of the devices can be found in the table below.

There are some obvious commonalities: All devices are designed around an accumulator based architecture, undeniably inspired from the Microchip PIC12 series. Interestingly, with only MDT as an exception, all vendors extended and modified their designs from the original. The reason for this is probably twofold: First, they want to avoid any legal issues with Microchip and secondly, the PIC12 itself is severely limited. Some of the major shortcomings are being addressed, such as lack of interrupts, addressable space of JMP/CALL, banking of memory/IO and severe lack of periphery.

Unfortunately, none of the vendors openly share details like instruction encoding or memory algorithms. Development for all device has to commence via vendor-provided IDEs. With exception of Holtek, all devices rely on high voltage programming interface and are not easily programming in-circuit. Only Padauk and Holtek offer devices that can be programmed more than once.

The „terrible“ 3 cent microcontroller

MCU	PTB150CSE	PMS150C-S08	BJ8P509F	PFS154-S08	MDT10P509(A)	HR7P153P45SA	PFS173-S08	HT68F001
Manufacturer	**Puolop** China www.puolop.com	**Padauk** Taiwan www.padauk.com.tw	**Bojuxing Industrial** China www.bjxmcu.com	**Padauk** Taiwan www.padauk.com.tw	**Yspring Tech / MDT** China www.yspringtech.com	**Eastsoft Micro** China www.essemi.com	**Padauk** Taiwan www.padauk.com.tw	**Holtek** Taiwan www.holtek.com
Package	SOP8-150mil	SOP8-150mil	SOP8-150mil	SOP8-150mil	SOP8-150mil	SOP8-150mil	SOP8-150mil	SOP8-150mil
Price (100pc)	0.0315 USD/pc	0.0334 USD/pc	0.0466 USD/pc	0.0692 USD/pc	0.0795 USD/pc	0.0811 USD/pc	0.0859 USD/pc	0.0875 USD/pc
CPU Architecture	PDK13 79 Instructions Stack in RAM Single cycle Interrupts: Yes	PDK13 79 Instructions Stack in RAM Single cycle Interrupts: Yes	BJ13 (PIC) 42 Instructions 5 level HW stack Single cycle? Interrupts: Yes	PDK14 82 Instructions Stack in RAM Single cycle Interrupts: Yes	PIC12 36 Instructions 2 level HW stack 4 clocks/instr. Interrupts: No	HR7P RISC 79 Instructions 8 level HW stack 2 clocks/instr.? Interrupts: Yes	PDK15 88 Instructions Stack in RAM Single cycle Interrupts: Yes	Holtek12 (PIC12) 63 Instructions 2 level HW stack 4 clocks/instr. Interrupts: Yes
ROM	1k x 13 bit OTP	1k x 13 bit OTP	1k x 13 bit OTP	2k x 14 bit Flash	1k x 12 bit OTP	2k x 16 bit OTP	3k x 15 bit Flash	0.5k x 12 bit Flash
RAM	64 Bytes	64 Bytes	49 Bytes	128 Bytes	41 Bytes	64 Bytes	256 Bytes	16 Bytes
Periphery	6 GPIO 16 bit timer 1x8 bit timer/PWM Analog comparator Brownout reset Watchdog timer	6 GPIO 16 bit timer 1x8 bit timer/PWM Analog comparator Brownout reset Watchdog timer	6 GPIO 1x8 bit timer Brownout reset Watchdog timer	6 GPIO 16 bit timer 2x8 bit timer/PWM 3x11 bit PWM Analog comparator Brownout reset Watchdog timer	6 GPIO 1x8 bit timer Brownout reset Watchdog timer	6 GPIO 2x8 bit timer/PWM 1x12bit ADC Brownout reset Watchdog timer	6 GPIO 16 bit timer 2x8 bit timer/PWM 3x11 bit PWM Analog comparator 1x8bit ADC Brownout reset Watchdog timer	6 GPIO 1x8 bit timer Brownout reset Watchdog timer
Clock	16 MHz internal 62 kHz internal	16 MHz internal 62 kHz internal	8 MHz internal External RC External XTAL	16 MHz internal 70 kHz internal External XTAL	8 MHz internal	16 MHz internal 36 kHz internal External XTAL	16 MHz internal 93 kHz internal External XTAL	- 32 kHz internal External XTAL
Programming	4+2 wire HV (12V)	4+2 wire HV (12V)	?	3+2 wire HV (8V)	?	3+2 wire HV (8.5V)	3+2 wire HV (8V)	2+2 wire ICP
Vendor IDE	?	Padauk Dev. Studio (ASM, Mini-C)	Bojuxing IDE (ASM)	Padauk Dev. Studio (ASM, Mini-C)	MDT-IDE, Yspring writer	?	Padauk Dev. Studio (ASM, Mini-C)	Holtek HT-IDE3000 (ASM)
OSS Toolchain	Tentative, SDCC >=3.9.0	Tentative, SDCC >=3.9.0	No	Yes, SDCC >=3.9.0	Yes, GPUTILS (PIC tools)	No	Yes, SDCC >=3.9.0	No
Notes	Website and datasheet in Chinese		Website in chinese, but english datasheet		PIC12C509 clone, horrible datasheet	Website and datasheet in chinese		Ultra low power

https://cpldcpu.wordpress.com/2019/08/12/the-terrible-3-cent-mcu/

Summary of my findings. PDF version here.

Individual Findings

Bojuxing Industrial

The BJ8P509F, priced at $0.0466, is a slightly enhanced version of the PIC12C509. The instruction set is extended from 12 bit to 13 bit. This allows for jmp instructions that can address the entire memory. In addition, interrupt capabilities and an extended HW stack were added.

An English datasheet is available and looks comprehensive enough to work with the device. Unfortunately, the vendor website and IDE documentation is only available in Chinese.

Eastsoft Micro

Eastsoft Micro has an extensive portfolio of PIC-derived microcontrollers. They call their flavor of the architecture “HR7P RISC”. It is a comprehensive accumulator based architecture with interrupt capability, 8 level stack and non-bankswitched access to memory and I/O.

There is one device in the sub $0.10 space available, the HR7P153P45SA. Notable features are the availability of a 12 Bit ADC, a low speed oscillator for lower power operation and the addition of two timers with PWM capability.

Unfortunately, both website and datasheets only seem to be available in Chinese.

Holtek

Holtek is a well-established microcontroller vendor from Taiwan. Their entry into this category, the HT68F001, is somewhat of an oddity: It’s a rather limited device with only 512 words of program flash and 16 bytes of RAM. The architecture is very similar to the PIC12 and can only be clocked from an internal 32 kHz oscillator. Since each instruction takes 4 cycles to execute, this results in only 8000 instructions per second! It appears that this device is targeting ultra-low power applications that have very low complexity requirements.

This MCU comes with excellent documentation. This includes their website, datasheets, application notes and IDE. It is also the only device to offer low voltage flash programming. Both of this sets Holtek somewhat apart from the rest of the field.

Given the limited functionality of their entry, however, it appears that the ultra-low-cost segment is not a priority for Holtek.

Padauk

It is very clear that the sub $0.10 MCU market is Padauks home turf. They have dozens of products in this price range, with a wide variety of features and package types.

All devices are based on Padauks MCU architecture, which is significantly extended over that of the PIC12: It uses separated I/O and SRAM memory regions and allows to address the full range without banking. In contrast to all other devices, the stack is memory mapped. Most instructions execute in a single cycle.

One interesting and very unique aspect is that Padauks architecture is geared toward synchronous multithreading, allowing to execute more than one program in parallel on the same MCU core using a time-slicing scheme. They call this concept “Field Programmable Processor Array” (FPPA). A similar concept is used in the XCore Architecture by XMOS. One useful application of multithreading in small MCUs is to create virtual periphery, e.g. UART, I²C, that is operated in parallel with the main program.

I picked three representative products in an SOP8 package: The PMS150C, the PFS154 and the PFS173. All of these only have single FPPA unit and therefore do not support multithreading.

The PMS150C is their lowest cost offer at $0.033, the original “3 cent MCU”. This device comes with 1 kiloword of one-time programmable memory and 64 bytes of RAM. The periphery is notably extended over many of the competing parts, offering a 16 Bit timer, an 8 Bit timer with PWM, LF oscillator and an analog comparator with 4 Bit reference voltage DAC that can be used to implement simple ADC functionality. All of this is sufficient to implement simple sensing and controlling functions.

The PFS154 comes at almost twice the cost. However, in contrast to the PMS150 it offers 2kW flash memory and can be programmed multiple times, which is much more convenient for actual development. The periphery has been extended with 3×11 bit PWM units, which look well suited to control RGB LEDs.

Finally, the PFS173 is an incremental improvement over the PFS154, adding an 8-bit ADC and extending flash to 3 kilowords and RAM to 256 bytes.

Padauk provides an IDE supporting development in Assembler and a somewhat cryptic dialect of C (“Mini-C”). They provide excellent datasheets in both English and Chinese as well as a bilingual website. Programming of the devices is accomplished by a 5 or 6 wire high voltage protocol, which makes in-circuit programming challenging.

Open source toolchain

Following the discussion on EEVblog, a small community has formed around the Padauk MCU with the goal of creating an open source toolchain for the device. Most of the activities are covered in this thread.

As of today (August 2019), reverse engineering the instruction encoding was completed, the programming protocol was documented, an open hardware programmer was developed and support for several flavors of the PDK architecture was integrated into SDCC. Development for all of the previously mentioned Padauk MCUs is now possible using a fully open toolchain.

Puolop

Puolop is a Shenzhen based Chinese provider of microcontrollers and various mixed signal circuits. They seem to offer a wide range of relabeled Padauk MCUs, specifically the older OTP version.

For example, the Puolop PTB150CSE appears to be identical to the Padauk PMS150C. Their pricing is slightly lower than the original ($0.0315 vs $0.334). It is not clear what the relation between Puolop and Padauk is, but it appears that Padauk is acting as a supplier to Puolop.

The company website and all documentation are only available in Chinese.

There does not seem to be any specific reason to consider Puolop MCUs over Padauks, other than saving fractions of cents on pricing.

Yspring Tech / MDT

Yspring Tech is a China based company that offers a wide range of devices that are functionally compatible to counterparts from Microchip. It appears that most of the product portfolio originated at MDT tech, which is a Taiwanese company and may have either been acquired or is on cooperation with Yspring. Microchip has taken issue with their business model in the past.

Ysprings addition to this review is the MDT10P509 which sells for $0.0795. This device seems to be an exact clone of the PIC12C509. While this could be useful as a low-cost replacement of the original product, it is clearly inferior to the other products in this category from a functional standpoint.

The MDT10P509 offers 1KW of OTP memory, 41 bytes of RAM and only a single 8 Bit timer as periphery. Like the PIC12C509 it offers no interrupts, only a 2 level HW stack and takes 4 clocks per instruction

Conclusions

Are these microcontrollers indeed “terrible”? That surely is a question of perspective. They address a specific category of low-cost, high volume, non-serviceable products with limited functionality. You need to wait for the push of a button and then let an LED flash exactly five times? You need to control a battery-operated night light? The sub $0.10 MCU is your friend to reduce BOM and shorten development time.

A caveat is that development for most of these devices is quite inconvenient due to limited availability of flash variants and lack of in-service programming. Debugging is usually only offered via in circuit emulators.

If you like working with low-end microcontrollers, the Padauk line-up is, without any doubt, the best choice. They offer the most powerful architecture, a wide range of devices including flash variants, good documentation, and are the only line-up with an open source toolchain.

Related

A LED-Candle based on the 3 cent MCU

In "candle"

Intelligent 7-Segment Display

In "Hardware"

Measuring the length of a WS2812 string

In "Hardware"

August 12, 2019 / Hardware, Padauk, PFS154 / 3 cent mcu, MCU, microcontroller, PFS154, PFS173, PMC150

19 thoughts on “The “terrible” 3 cent MCU – a short survey of sub $0.10 microcontrollers.”

Jay Carlson

August 14, 2019 at 2:46 am

Great write-up! Just how close are the ISAs to the PIC stuff? I wonder if the assembler output of XC8-compiled C code could be translated to any of these parts?

cpldcpu

August 14, 2019 at 5:44 am

Thank you! Your article was obviously a great inspiration – and totally unreachable from a point of comprehensiveness.

I don’t think it will be easy to retarged XC8. Firstly, none of these vendors published their instruction encoding. It had to be figured out from the IDE for the Padauk. Secondly, there is a huge variety in ISAs. Some MCUs like MDT and BJX appear to be quite close to the orignal, but others divert heavily from the PIC architecture. For example by splitting I/O and memory into two regions, accessed by disparate instructions.

However, SDCC works for Padauk and could possibly be also retargted to other devices, once they become more accessible.

Dave Jones (@eevblog)

August 14, 2019 at 3:02 am

Great article!
I’ll be trying the Padauk open source compiler and programmer soon I hope, sounds very promising.

cpldcpu

August 14, 2019 at 5:45 am

Thank you! You should give the 3 cent MCUs more love, a lot has happened in the mean time.

---

gp

August 14, 2019 at 3:56 am

Great article, thank you for posting this research. Are any of these architectures close to the hardware requirements for a barebones Linux port?

---

cpldcpu

August 14, 2019 at 5:46 am

Well, you could probably print the first two lines of the boot message…

Pingback: Padauk PMS150C "3 Cents" MCU Supports SDCC Open Source Toolchain

---

C. Barry Ward

August 14, 2019 at 2:27 pm

Thanks for the great reminder that there other microcontroller vendors out there. The table is great. As someone that’s been putting microcontrollers (PICs and others) in consumer products for decades it will be great fun to try some of these out.

As to the lack of reprogrammability, is it really an issue? At three cents each, three bucks gets you 100 iterations… As to the toolchain, an assembler/linker is likely all you need. If you really want to get your three cents worth you’re probably going to be programming this thing in assembly anyway.

Pingback: Bill and Dave's Excellent Equipment | The Amp Hour Electronics Podcast

Cluso99

August 28, 2019 at 11:02 pm

Is there a distributor for Padauk or can you buy direct (in reels)?

cpldcpu

August 28, 2019 at 9:04 pm

Well, there is LCSC.com. They also list some distributors on their website.

zoobab

August 29, 2019 at 12:55 pm

I am wondering if this custom flasher could not be replaced with a simple arduino.

What makes it so special, is there some special voltage?

cpldcpu

August 31, 2019 at 8:38 am

Yes, it requires very accurate voltage control of VDD and programming voltage.

Eric

August 29, 2019 at 1:14 pm

Hi all,

A few weeks ago, I created a little board with a PADAUK PMC234, a $0.30 processor. It works fine a very easy to use. I used the official In-Circuit emulator and the official programmer. Both works great and about $120 each at LCSC.
http://synthelectro-fr.blogspot.com/2019/04/sram-select-command-ver-10.html

I don’t know if the open hardware programmer can program this PMC234. If anyone can tell me about it…

spth

October 7, 2019 at 1:03 pm

Currently, the only dualcore Padauk devices supported by the free programmer are the PMC251 and PMS271.
However, I assume that the hardware could support the PMC234, and only the software would need to be updated. You can file a request for support for those devices at
https://github.com/free-pdk/easy-pdk-programmer-software/issues

E

November 25, 2019 at 12:58 pm

Very interesting write up. I have been following articles on the cheap mcu’s for a while. Time to give it a try. The padauk line is the best to give a try

diakoakhavian

December 20, 2019 at 4:59 am

Hi, Jay
I’m a new to the world of embedded programming and I have a posted a question on stack overflow which was downgraded. I was hoping if you could help me with a Micro-controller that I’m currently working on for my internship and feel lost. It made by a Chinese company called EastSoft. It interacts with the PC using a hardware programmer and there is an LCD attached to the main board . I have managed to translate the datasheet and was wondering if you could help me make sense of it. Some of my questions include:
1. What is the format of the hex inside the MCU?
2. Identifying the process
3. Identifying the set architecture
The main board drives 2 water sensor and a LCD. The default maximum temperature set by the manufacturer is 70 degree Celsius. I just want change this default maximum to temperature to 69. Can this be done by reverse engineering such as accessing the debug ports and changing the temperature parameter in the firmware or I have the flash the firmware and write the whole program myself ?

cpldcpu

December 30, 2019 at 9:40 am

I am not Jay, but let me respond nevertheless.

Your best bet would probably be to contact the manufacturer and ask for more information. I have not spent too much time on the Eastsoft devices, but it looks like they are concentrating on marketing only the China. Hence there is no non-chinese documentation.

Your questions:

1. What is the format of the hex inside the MCU?
2. Identifying the process
3. Identifying the set architecture

Especially for more obscure MCUs this is something that the manufacturers often do not share openly. If you really need this information, one way would be to deduce it from the toolchain and code examples. This takes a lot of time and is probably only worth it of you like that challenge.

Otherwise, simply switching to a different MCU may be the better option.

Kaz

February 6, 2020 at 12:18 pm

I was just wondering if anyone has tried using the MDT10P509. I had a project that used PIC12C508/12F508, and wanted to see if this MDT10P509 is a viable alternative, but my various PIC programmers don’t seem to want to program this part. To be honest, I haven’t used my non-ICSP programmers in years, so I don’t know if they’ve gone bad in the meantime, or if some different programmer is needed for the MDT parts. I can’t seem to find that information. Just wondering if anyone else has looked at these and had different results.