Microprocessor systems EMISY

Basic microcontroller peripheral circuits

Semester 20L – Summer 2020

© Maciej Urbanski, MSc

email: M.Urbanski@elka.pw.edu.pl

WUT ise

Important remark

This material is intended to be used by the students during the Microprocessor Systems course for educational purposes only. The course is conducted in the Faculty of Electronics, Warsaw University of Technology.

The use of this material in any other purpose than education is prohibited.

This material has been prepared based on many sources, considered by the author as valueable, however it is possible that the material contains errors and misstatements.

The author takes no responsibility for the usage of this material and any potential losses this usage can lead to. Furthermore the author will be very grateful for pointing out any errors found and also for any other useful remarks on the course material and potential upgrades.

WUT ise

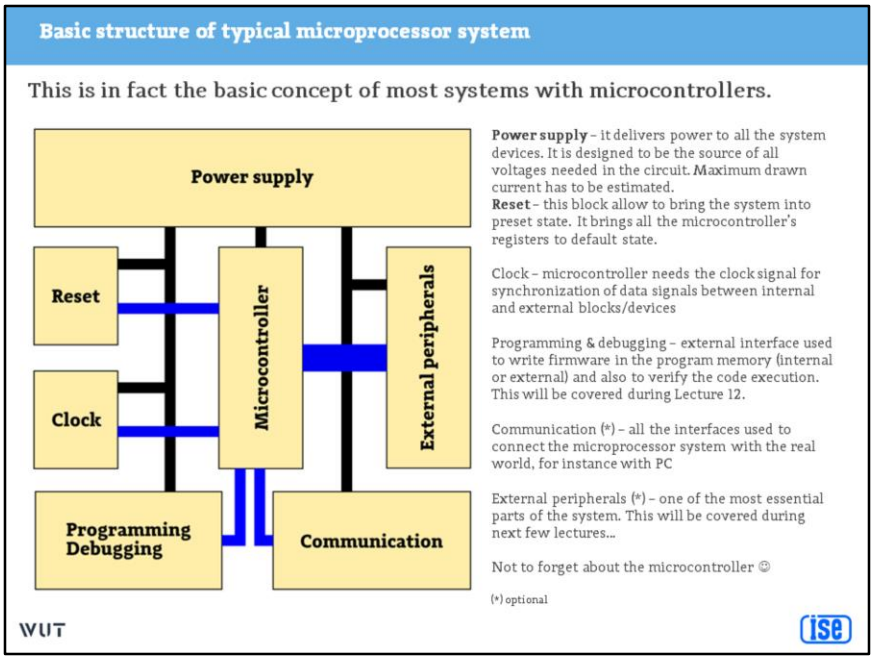
How will these lectures look like?

I've tried to record something for you, but the result was not acceptable, due to many quality reasons.

That is why I decided to upgrade my lecture slides with some extended description and text comments.

After each important slide I placed some more detailed text description.

WUT



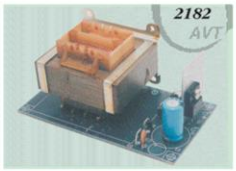
In the upper slide there is a simplified block diagram of a typical microprocessor system. As everything in electronics (or almost everything) such system requires power supply that will create proper voltages for all other sections of the system. There must be a microcontroller in such a system that will drive everything – you may consider it as a master of this system. It is supplied, like all other modules, via power buses (black bars in the drawing) and it communicates with the modules via communication buses (blue bars). In most cases microcontroller requires also some resetting circuit that will ensure that the code execution is OK and in case of anything wrong will reset the MCU to its default state. MCU needs a clock source as well as it is a synchronous device. It communicates with external peripherals, like for instance by using GPIO to drive LEDs, fans, relays, etc. For some more sophisticated peripherals (like for instance analog to digital converters – ADCs) it uses communication interfaces, like I2C, SPI, USB, Ethernet, etc.
Optionally (but in case of prototype circuits it is required) such system may be equipped with programming and debugging interface, so you as a designer are able to modify your firmware and check how it is executed.

Power supply

It is believed that the electronic circuit works better when plugged into power supply...

Power supply is a device that converts the external power source parameters to fit the designed circuit requirements. These requirements are:

- Number of outputs (separated or not)
- Voltage levels for the outputs
- Current limits for the outputs



Simple power supply using a transformer

Commercial switching power supply

There are basically two types of power supplies:

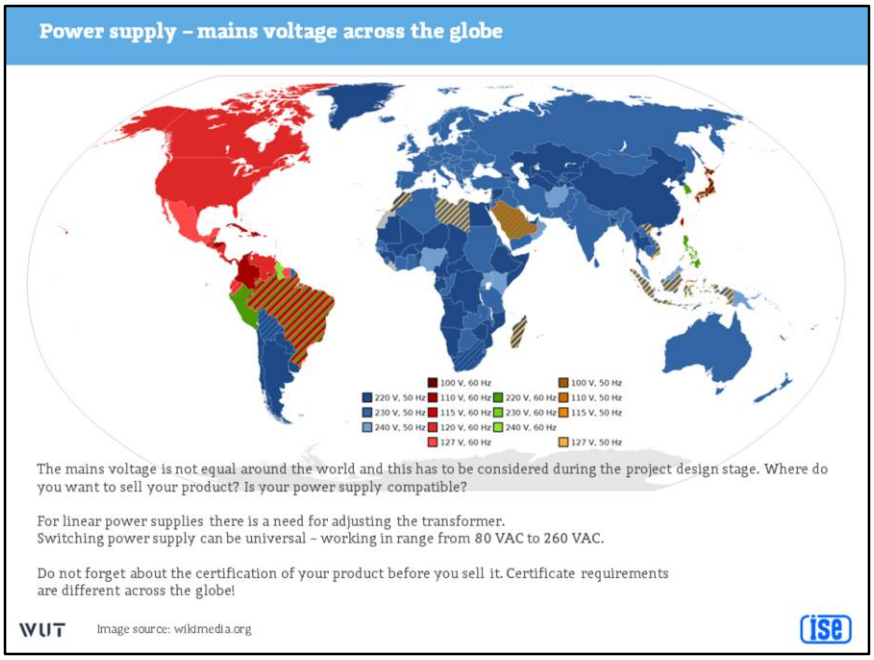
- Linear power supplies
  - Low noise
  - Low efficiency – high thermal losses
- Switching power supplies
  - High efficiency
  - May be the source of high frequency noise in circuit

WUT Image source: elportal.pl, farnell.com ise

When it comes to power supplies – it is a device that converts one Energy source to other. In most cases it takes some external Energy source, like for instance wall socket AC source and converts it to a proper DC source for designed system. Such conversion requires that the output voltages are within the needs of the system, as well as current efficiency of each of the outputs.

There are in principle two types of power supplies – switching and linear PSUs. Both of them have their pros and cons and both of them are used in different applications. Linear power supplies are usually low noise, which is why they are commonly used in precise analog electronics and measurement devices. On the other hand they are usually also big and unefficient – this means that they convert too much of input power to heat instead of output power – you do not want that. An example of such power supply is a simple AC transformer power supply, shown in the upper picture.

Switching power supplies, on the contrary, are usually very efficient (even up to 95%) but unlike linear power supplies they are noisy and therefore their usage in precision analog circuits is risky. They are also usually very compact in size (like the module in the bottom picture).



When it comes to different types of power supplies – you must know that the mains voltage around the world is not constant. For instance in Europe we use 230V AC – this means that the voltage in European wall sockets is a sinewave with RMS (root mean square) value of 230V and frequency of 50Hz.

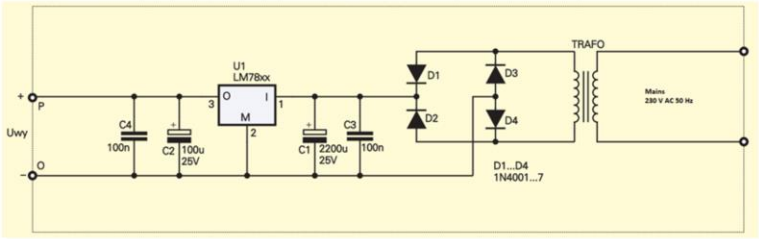
Now, if you design a circuit and want to sell it (you want to make money, right?) then you must remember that fact. Your circuit's power supply must be compatible with the AC voltage in the country you want to sell it in.

In most cases linear power supplies require modifications as they are in most cases designed for a specific AC voltage range. There are ways to make them more universal, like making some special modifications in the AC transformers.

The other way to solve the problem is to use a switching power supply – in most cases their input voltage range is so big that it covers the whole spectrum of voltages around the world.

Power supply

Transformer based power supply



Transformer (TRAFO) converts input 230V AC sinewave 50 Hz into low voltage sinewave 50 Hz.
The low voltage sinewave signal is then rectified with the full bridge rectifier (D1-D4).
Capacitors C1 and C3 filters rectifier ripple voltage.
U1 is linear voltage regulator. It converts DC rectifier voltage to required DC voltage (for example +5V)
Capacitors C2 and C4 filters output DC voltage.

WUT Image source: Elektronika dla Wszystkich, 2/1998 ISE

Here we have an example of a simple transformer based linear power supply. Starting from the right – AC transformer converts AC high voltage (230V) to low voltage AC (let's say 12V in the example). Then such low voltage AC is rectified by the diode bridge and the result is quasi DC signal (rectified sinewave). In order to feed it to the voltage regulator (U1) we need to filter and smooth it – for the purpose we use C1 and C3 capacitors. Notice that the C1 capacitance is very big as it must act as a source for the time when the rectified sinewave comes down to 0V.

U1 is an ordinary linear voltage regulator. Its job is to convert DC voltage to DC voltage and that is what it does. In the example it is an LM78xx used – these are very common positive voltage 1A regulator, with fixed output voltage, available in many different output voltage variants. In our example lets assume that we have LM7805 – a typical 1A +5V regulator.

Caps C2 and C4 are filtering output DC voltage.

Everything should be OK, except it is not. The answer is in slide 9.

https://en.wikipedia.org/wiki/Diode_bridge

Linear voltage regulators – fixed and adjustable

Voltage regulator is a device that converts input DC voltage to output DC voltage.

Voltage regulator has to:
- Ensure proper output voltage stability in required voltage and current range
- Ensure proper output current limit
- Be invulnerable to output short
- Have the highest possible efficiency – in other words it should not generate too much heat
- Separate output from input noise and ripple

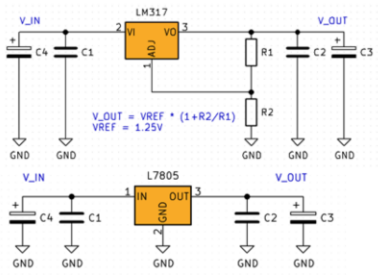
There are many different voltage regulator available. Usually they have fixed or adjustable output voltage.

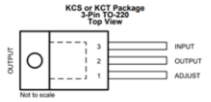
In typical applications the most popular voltage regulators are classic LM78XX, LM79XX, LM317, LM337. These are fixed positive (78), fixed negative (79), adjustable positive (317) and adjustable negative (337). They are available in standard output voltages: +5V, +3,3V, and adjustable in range from 1,3 to 30 V.

They are old and robust devices, however not recommended for modern demanding applications.

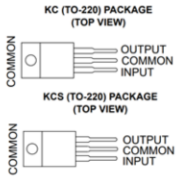
WUT ise

Linear voltage regulators – fixed and adjustable





LM317 pinout (THT TO-220)



LM78XX pinout (THT TO-220)

Capacitors and resitors values should be selected according to the datasheets and application – capacitors should provide required output ripple level, depending from output current.

Limitations:

- Input voltage – not higher than +35V DC
- Output current – not higher than 1.5 A (may differ between manufacturers and packages)
- **Maximum power dissipated** –

$$P_{TOT} = (V_{IN} - V_{OUT}) \cdot I_{OUT}$$

- The voltage drop across the regulator will be the main source of heat

WUT

ISE

This slide presents typical applications of low power linear voltage regulators. The examples given here are old, robust and commonly used voltage regulators.
You may use them in your project applications, but when designing something professional I rather recommend to look for something more modern.

There are two applications here – LM317 and LM7805 (already described in slide 7). This is a typical application of LM317 – input and output capacitors, just like in LM7805 application, but there is something strange, called a voltage divider.

https://en.wikipedia.org/wiki/Voltage_divider

It is used to take a fraction of the output voltage and give it back to the regulator so it knows what is the output voltage. According to this information the LM317 will update its output voltage value. By adjusting R1 and R2 it is possible to adjust the output voltage range.

When designing voltage regulator circuits – you must remember about some limitations. The input voltage must not exceed +35V or +30V (for the mentioned regulators). The output current should not exceed 1.5A and what is most important –

the maximum power dissipated must not exceed the limits, otherwise the regulator will burn.

The limitations are available in the datasheets for every voltage regulator and you must read it (Absolute Maximum Ratings).
http://www.ti.com/lit/ds/slvs044x/slvs044x.pdf
http://www.ti.com/lit/ds/symlink/lm340.pdf

This will be a part of the project to design some simple voltage regulators – either linear or switching. In most cases this will be simply using the basic application diagram from the datasheet.

Switching voltage regulators – solution for all the problems?

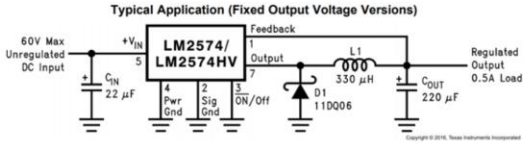
In switching voltage regulators the active element (transistor) is not driven all the time – the driving signal is switched. By proper setting of switching time (time on and off) the amount of charge transferred to the output is adjusted – voltage regulation.

Switching voltage regulators are significantly more efficient than linear regulators (efficiency up to 90%) – they usually do not require big heatsinks (if at all).

Switching mode regulators are based on pulse-width modulation (PWM) to control the average value of the output voltage.

Their application circuit is a bit more complicated than the linear regulator application.

They can be used to generate lower, higher or inverted output voltage, compared to input voltage.



Typical application of LM2574 fixed switching voltage regulator.

WUT Image source: http://www.ti.com/lit/ds/symlink/lm2574.pdf ise

This is an example of a switching voltage regulator. The circuit is quite similar to previous ones, except there are some additional elements here. The LM2574 is a compact switching regulator. If it turns out that you need a switching regulator in your project then first please consult this with the project manager.

When to use switching voltage regulator and when linear one?

Linear regulators are better for low noise and precision applications, for instance for analog signal conditioning, operational amplifiers circuitry, audio preamplifiers, high resolution ADCs, etc.

Linear regulators are best when it is required to quickly react on input voltage change. It is achieved due to analog negative feedback loop.

Linear regulators are simpler and usually cheaper in low power applications.

Switching regulators are best when high efficiency is required – battery powered devices, low power devices.

Switching regulators are usually significantly smaller then linear regulators (with same max. output current).

In high power applications switching regulators are cheaper than linear regulators.

Switching regulators are usually not suitable for low noise applications.

Sample power supply scheme for high precision microprocessor system

This is sample system supplied from +15V, -15V and +7V DC (coming from custom switching power supply).

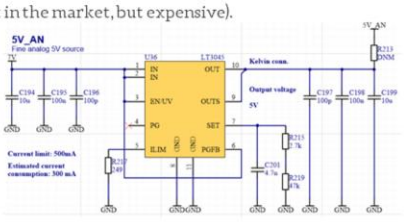
This power supply circuit has to provide +5V for analog circuits, +14V and -3V for operational amplifiers, +3.3V for digital circuits.

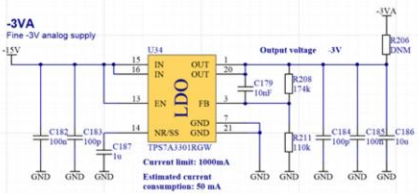
Low noise analog regulator is LT3045 (one of the best in the market, but expensive).

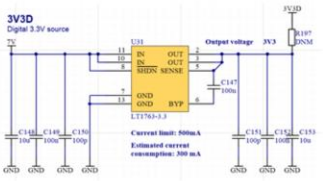
The negative regulator is TPS7A3301, also low noise.

For digital 3.3V LT1763 in fixed 3.3V version was chosen.

Estimated current consumption made it possible to use only linear regulators.







WUT ise

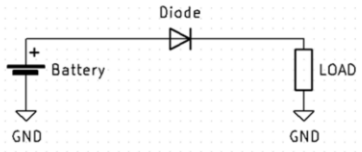
In the upper slide there are examples of modern linear voltage regulators, designed for a precise RF monitoring circuit. The circuit required -3V, +14V and +5V and for each of them a separate regulator is used. The parts used in these examples are modern voltage regulators that I recommend for precise applications. They are not cheap though.

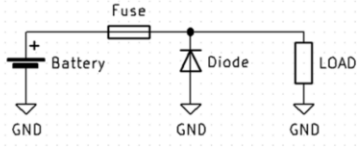
Battery supply

There are many ways to connect battery to the circuit.

Good advice. If something stupid is possible to be done in your circuit then you can be 100% sure that there is at least one user of your circuit that will do it!

Always make your circuit "user-proof".



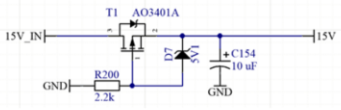


Diode is used to prevent LOAD (your circuit) from reverse polarity connection of the battery.

The drawback is that the voltage across the load is one diode voltage drop less than battery voltage.

Destructive protection. When the battery is connected the wrong way it causes the very high current to flow through the fuse and the diode. This very high current will blow the fuse and thus prevent the load, according to Kirchoff's laws.

There are more clever ways of protecting the device from reverse battery (or power supply) connection.



WUT

ISE

The MOSFET example:

https://hackaday.com/2011/12/06/reverse-voltage-protection-with-a-p-fet/
https://www.infineon.com/dgdl/Reverse-Batery-Protection-Rev2.pdf?fileId=db3a304412b407950112b41887722615

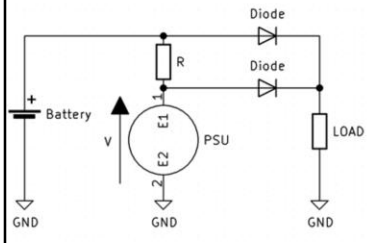
Emergency battery power supply connection

Sometimes there is a need to make your system invulnerable to power supply loss, like blackouts, electricity failures, etc.

This often happens in clock circuits. It would be nice to have a clock that will hold the time information even with the power supply turned off.

The device should detect that there is a problem with power supply and switch to power saving mode (Lecture 11). The battery should then supply power only to the Real Time Clock module (RTC) or selected blocks of microcontroller.

According to laws of physics you MUST NOT connect two voltage sources in parallel!
According to laws of physics you MUST NOT connect two current sources in series!



This is a simple way to connect emergency power supply to the load. The PSU voltage must be higher than battery voltage (and usually it is). When PSU is on, the upper diode is reverse biased and does not allow current to flow. The bottom diode is forward biased.
When the PSU is off the bottom diode is reverse biased (or not biased at all)

The diode circuit creates OR gate.
The resistor R is optional and needed when there is a need to charge battery when it is not used.

WUT

ISE

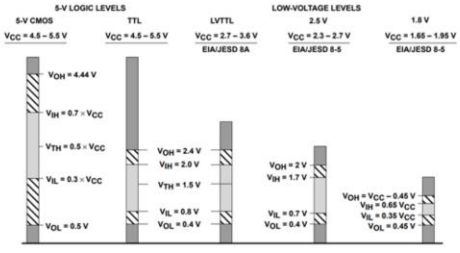
Typical voltage levels for integrated circuits in microprocessor systems

First digital IC chips were designed to work with +5V DC power supply.
With the rapid development of electronics and digital logic, lower voltage standards were introduced.

The increase of complexity of integrated digital circuits forced the designers to reduce size of transistors in semiconductor designs. This caused technological layers, like oxidation layers for insulating transistors, to become thinner and thinner, thus more vulnerable to electrostatic field strenght. The field strenght is proportional to the supply voltage. This is why making complex digital chips forces to use lower voltage levels.

More can be read in:
http://www.tij.co.jp/jp/lit/an/sdaa011a/sdaa011a.pdf

3.3 V standard is used in various modern microcontrollers (STM32, AVR)
2.5 V and 1.8 V standards are used mostly in high speed digital circuits, featuring FPGAs, fast digital differential buses and fast ADCs
5V standard is still used in older digital circuits, but also in applications exposed to strong electromagnetic interference (EMI), for instance in industry applications.

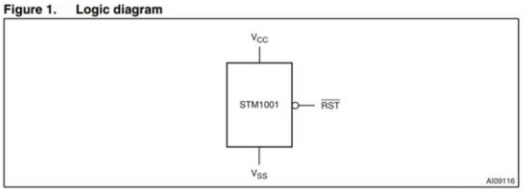


WUT Image source: http://www.tij.co.jp/jp/lit/an/sdaa011a/sdaa011a.pdf ISE

Power supply monitors and reset drivers

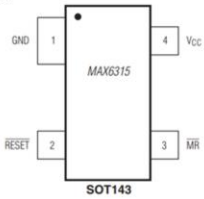
The STM1001 microprocessor reset circuit is a low-power supervisory device used to monitor power supplies. It performs a single function: asserting a reset signal whenever the V_{CC} supply voltage drops below a preset value and keeping it asserted until V_{CC} has risen above the preset threshold for a minimum period of time (t_{rec}).

Figure 1. Logic diagram



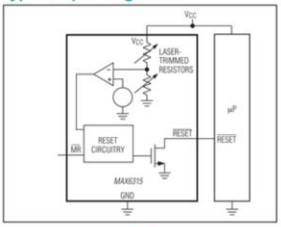
SOT23-3 (WX)

TOP VIEW

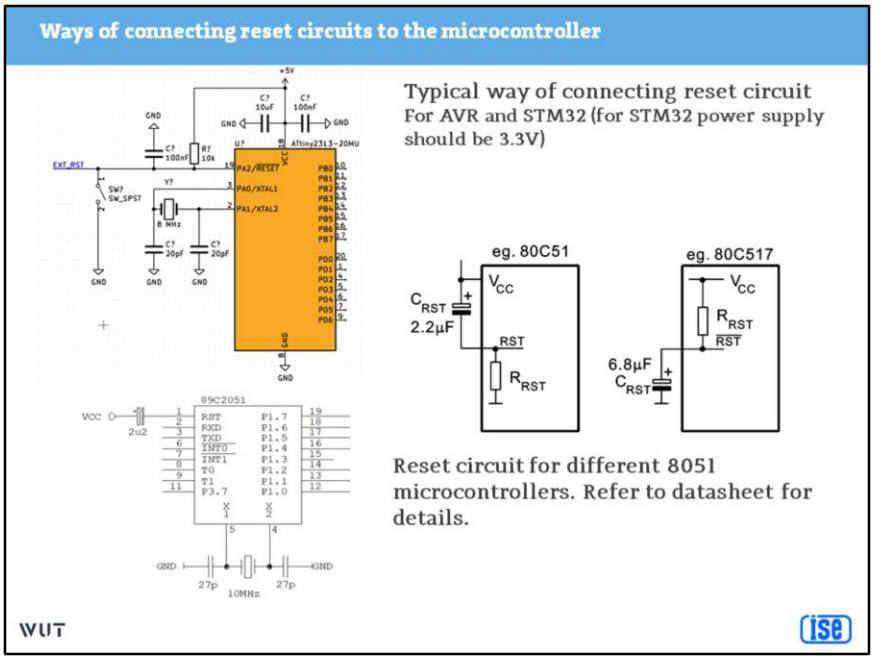


SOT143

Typical Operating Circuit



Now let's move on to reset circuits. First we will discuss so called reset drivers. The chip given above is an example of a reset monitor chip.
In this case it monitors the power supply voltage and if it drops below limit the chip resets the microcontroller and keeps it in the reset state as long as the power supply is not correct. This is a proper way – if the power supply drops and we do not reset the MCU it's behaviour is unpredictible.



There are many different ways to connect reset circuits to the MCUs and it must be emphasized – the circuit depends on the microcontroller, so always check the datasheet first.

In the slide above there are most common examples for AVR and STM32 and 8051. For AVR and STM32 the most common way to connect reset circuit is as follows: pull-up resistor (around 10k) that will keep the weak logic one on the reset input, and a 100nF decoupling capacitor that will form a low pass RC filter, to filter any spikes or noise. In case there is a need to reset (by user, or monitoring circuit) it is connected to EXT_RST. By shorting this node to ground a strong low level is forced at the MCU input and the status is reset.

Unlike STM32 the 8051 scheme is different and what is the worst – **it may differ between different manufacturers.** The most common approach for 8051 is a 2.2uF capacitor connected between reset pin and VCC. This chip is reset by shorting RST pin to VCC.

In reset state, all 8051 pins are in logic 1 state.

Clock circuits

Every synchronous digital device, like microcontroller, requires synchronization signal for proper work. This signal is called clock signal and is used to manage data flow between microcontroller blocks and external peripherals. In most cases it is square wave or sine wave signal.

Clock signal determines the microcontroller working speed and current consumption.

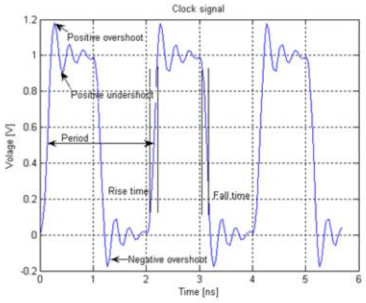
Clock signal frequency usually must be chosen carefully, to fit external peripherals used in the system.

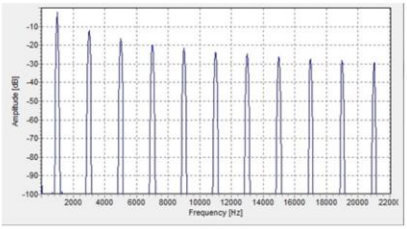
It is often required in the code to generate precision time delays. For the purpose the proper clock frequency can be very useful – especially if the frequency can be divided by 2^n, where n is positive integer value.

Maximum clock frequency is limited by the type of microcontroller. For instance AVR microcontrollers can work with clock signal frequency up to 20 MHz and STM32 with input clock signal frequencies (to be more precise – crystal resonator frequencies) up to 48 MHz. STM32 can use internal PLL to work with clock frequency up to 172 MHz, depending on family (F0 to F7).

Clock signal – ideal square wave versus real square wave

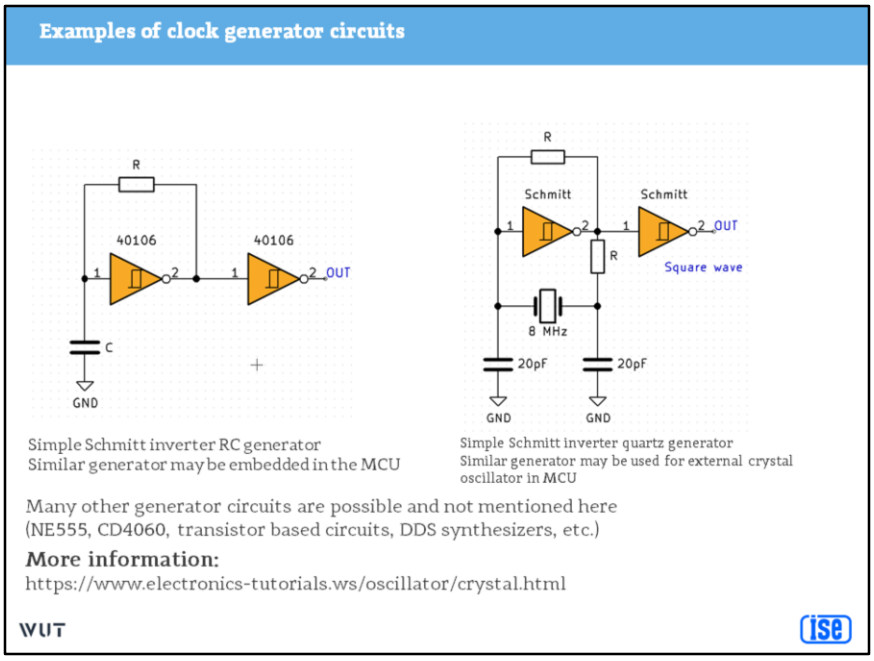
The world is not perfect. Due to signal integrity issues, mainly impedance mismatch, the square wave is significantly deformed by overshoots and undershoots. These effects are particularly annoying when working with high frequency signals – in such cases careful PCB design and signal integrity simulations have to be done. If this cannot be done, then it should be considered to reduce rise and fall times of the clock signal (usually by a series resistor)





The square wave signal can also be a source of high frequency EMI – electromagnetic interference – it can affect other analog signals in the system (or worse, in some other system), causing malfunctions.

WUT Image source: wikipedia.org ise

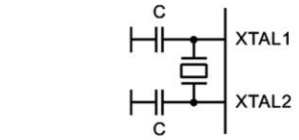


The slide presents some basic clock generator circuits. The left one is a simple RC generator, using one (second is just a buffer) Schmitt inwerter. Let's assume that there is no charge in C –the voltage across it is zero. The output of the first inverter is logic one (some positive voltage). Via resitor R the capacitor is charged – the voltage across it rises slowly. When this voltage is high enough to be considered a logic one the inverter changes its state and the capacitor starts to be discharged via R resistance. The charging/decharging speed defines the frequency. It must be emphasized that the circuit is very simple and not very precise nor stable and should not be used as a precise clock source.

The right picture presents a crystal resonator based clock circuit. Capacitors below the 8MHz crystal resonator are used to force the crystal resonance at a specified frequency (8 MHz in this case).

Clock signal oscillators in the microcontrollers – crystal resonator based

A quartz crystal resonator is a piezoelectric device transforming electric energy into mechanical energy and vice versa. The transformation occurs at the resonant frequency (or its harmonic).



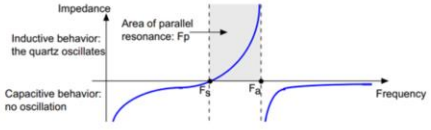
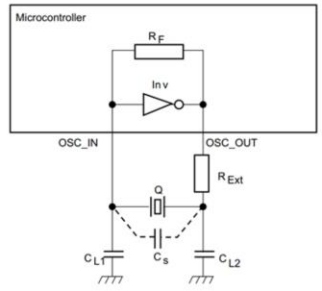
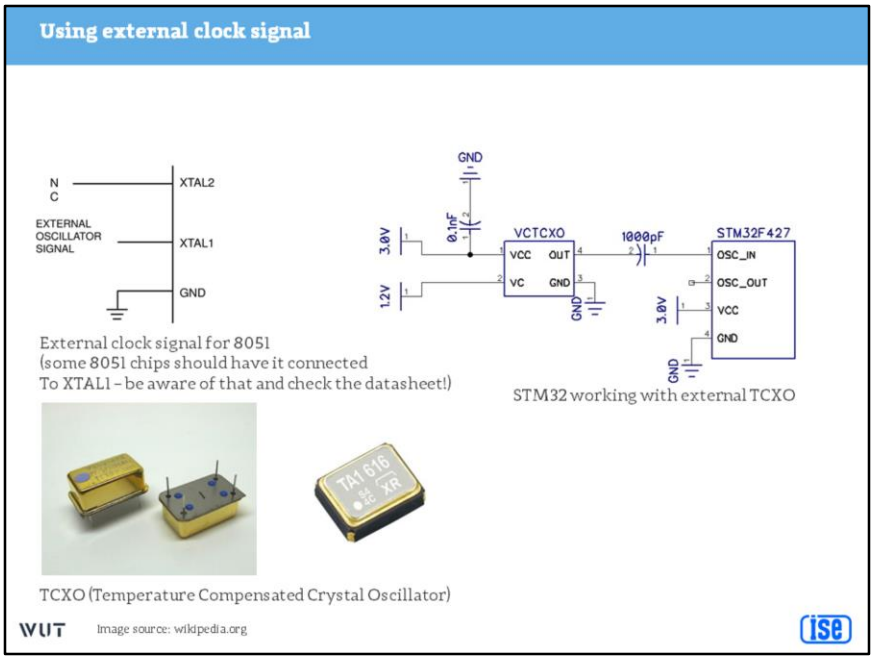


Figure 5. Pierce oscillator circuitry

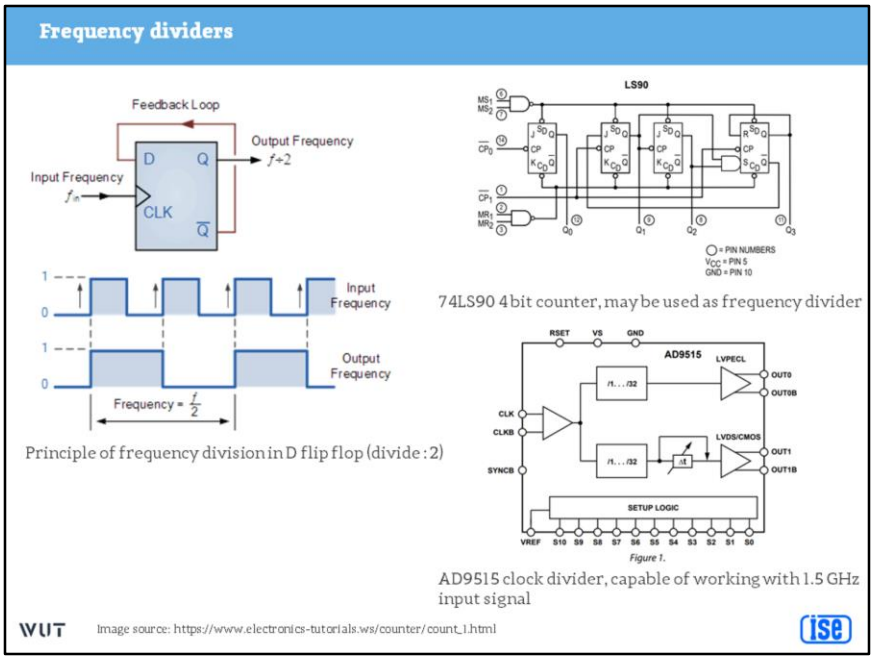


Values of CL1 and CL2 capacitors should be selected according to required quartz oscillation frequency and microcontroller specification. Refer to ST note AN2867 for details and equations. Usually CL1=CL2 = between 10-20 pF (not a rule!!)

WUT Image source: ST Application Note AN2867 ise



It is also possible to use external clock source. In most cases this will be a square/sine/differential wave. Let's assume the most common situation – the square wave. Such signal (if it meets voltage requirements for the MCU) can be supplied directly to a proper XTAL (crystal resonator) pin. Such signal can be taken from some other chip, other microcontroller or FPGA, or from dedicated crystal **oscillator**. It is not the same as crystal **resonator**. Crystal oscillator is a complete signal generator with fixed and stabilized output frequency. It should be used as a clock source in precise applications, like for instance frequency counter/meters, etc.



It is not always possible to get a proper frequency for your application. Let's assume that the MCU in the design can work with input frequencies not bigger that 25 MHz. The clock source is defined at 50 MHz (quite common). In order to create a proper (and synchronized with 50 MHz) clock signal at lower frequency a frequency divider is needed. The simplest frequency divider is a D flip flop in configuration as shown above. Single D flip flop cell can divide frequency by 2. In case some other ratio is needed an n-bit counter may be used for the purpose.

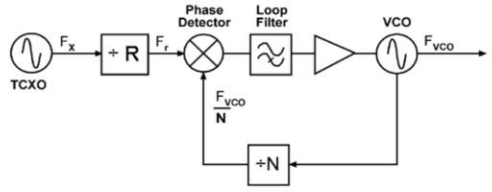
It is also worth to mention that there are some other ways to deal with clock signal. In case the clock source is an RF source and the system must be synchronized with it one may use a dedicated RF clock divider, like for instance AD9515.

Frequency multiplier - PLL

The PLL circuit performs frequency multiplication via comparing phase difference between reference signal (TCXO) and signal from frequency divider (:N). The negative feedback loop will try to drive the VCO (Voltage Controlled Oscillator) in such way to achieve $F_{VCO} = NF_r$. This way the phase detector is driven with two signals of same frequency – this is a stable state, we say that the phase loop is **locked.** As can be seen the circuit allows for integer-n multiplication of input frequency.

In digital systems the VCO is replaced with DCO (Digital Controlled Oscillator) and the output signal is square wave.

Integer-N (classical) PLL Block Diagram

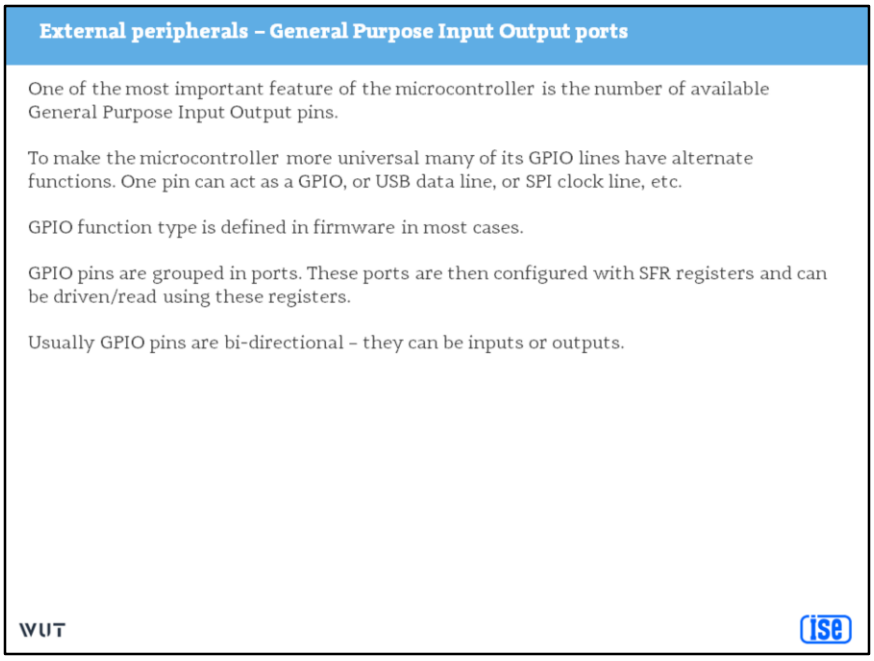


Such devices are embedded in modern microcontrollers, like STM32 series.

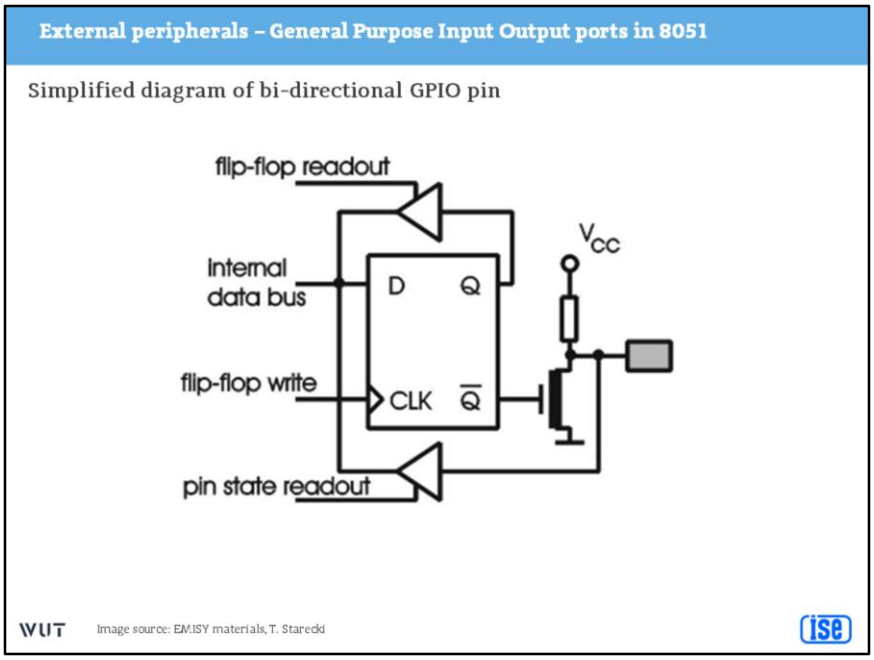
WUT Image source: http://www.ti.com/lit/an/swra029/swra029.pdf ISE

And what if we need a higher frequency that a source can deliver? In such case a PLL circuit is needed. PLL- Phase Locked Loop – is a circuit designed to alter the frequency and keep it in sync with the input. It is done as follows.

The input TCXO creates a signal with fundamental frequency of Fx. It is initially divided (freq) and fed into Phase detector. Phase detector drives the other oscillator – so called voltage controlled oscillator (VCO). Phase detector will do anything it can to make sure that the frequencies at its inputs are equal.
Let's assume that we have some output signal from the phase detector. After filtering and amplifying it goes to the VCO. VCO creates the output signal and fraction of this signal goes back to the phase detector **via the frequency divider.** This means that in order to keep the frequencies at the inputs of the phase detector equal the VCO frequency must be N Times higher that the Fr frequency. And that is how it works. By adjusting the N ratio the output frequency can be multipled N Times.



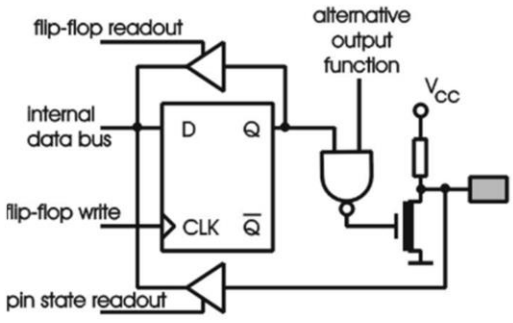
Now let's talk about the „hands" of the MCU – the General Purpose Input Output Pins. They are used to communicate with the world. They are used to drive various things, like LEDs, motors, fans, relays, switches, lights, etc.



The basic structure of 8051 GPIO pin (single pin) is shown in the upper slide. As everything in MCU, GPIO pins are supplied with clock signal and they are updated according to this clock. The data is fed to the data input of the flip flop and the output is updated according to the clock. The output status can be read using flip-flop readout and the pin state can be read using pin state readout.

External peripherals – General Purpose Input Output ports in 8051

Simplified schematic of GPIO pin with alternate function for the output



WUT Image source: EMISY materials, T. Starecki ise

Many GPIO pins have some alternative functions, like I2C or SPI communication pins. The way how it's done in hardware is shown in the upper figure. This is an example of an alternative output function, using a NAND gate.

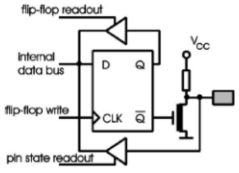
External peripherals – General Purpose Input Output ports in 8051

Read-modify-write instructions

GPIO read operation can be done in two ways:
- By reading port state register
- By reading input line state

Instructions that reads the state registers are called read-modify-write instructions.

- ANL P2, A
- ORL P1, #10
- XRL P3, A
- JBC P1.3, next
- CPL P1.0
- INC P2
- DEC P3
- DJNZ P2, alpha
- MOV P*x.y*, C (eg. MOV P1.0, C)
- CLR P*x.y* (eg. CLR P1.1)
- SETB P*x.y* (eg. SETB P2.3)



WUT Image source: EMISY materials, T. Starecki ise

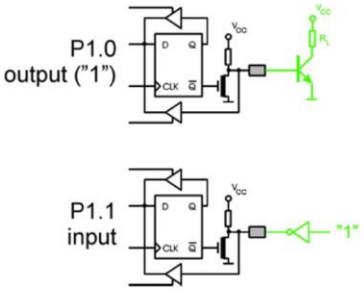
External peripherals – General Purpose Input Output ports in 8051

Read-modify-write instructions

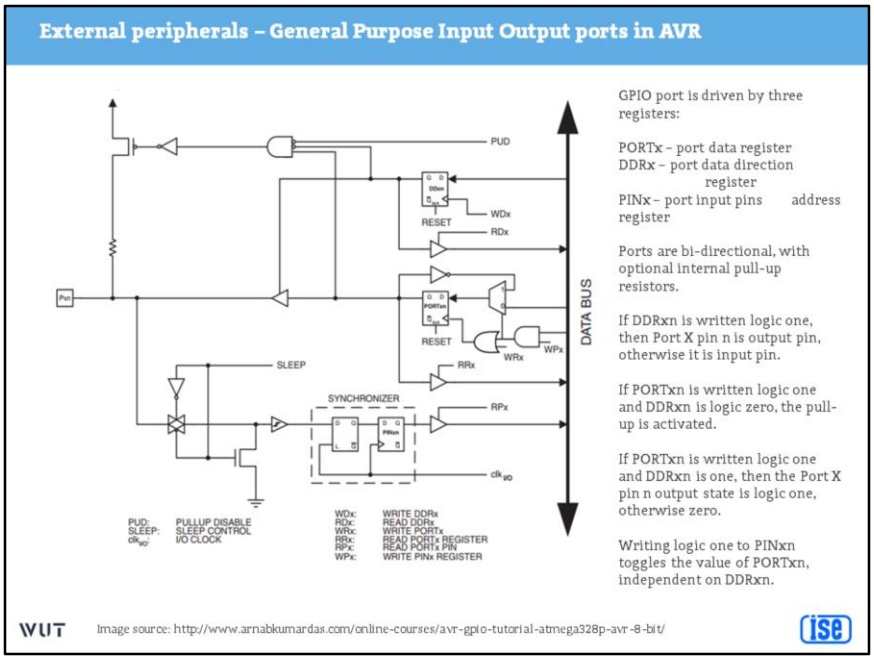
In some cases the unawareness may lead to read errors – instead of reading status register the current state of input lines may be read.

read/write instructions:
MOV A, P1
ANL A, #0FFH
MOV P1, A

read-modify-write instruction:
ANL P1, #0FFH



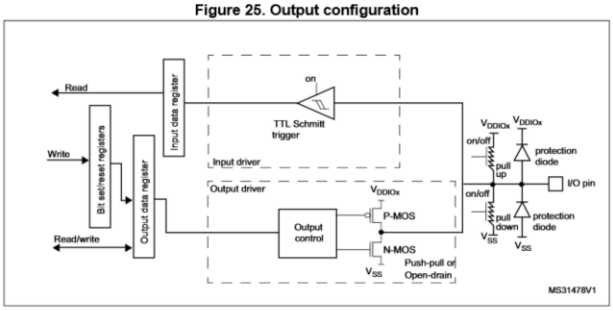
WUT Image source: EMISY materials, T. Starecki ise



This is an example of an AVR microcontroller GPIO pin. It is driven by three registers, as described. We can see some other additional features, like internal pull-up resistor that can be disconnected, or some additional reset and watchdog circuits, used when the MCU is in reset stated or being turned into reset state (by the watchdog for instance). We will talk about watchdogs in further lectures. Right now all you need to know is that a watchdog is a counter that should be periodically zeroed by the MCU. If not and the watchdog counter is full, then it resets the MCU. It becomes full if something wrong with the code execution happened.

External peripherals – General Purpose Input Output ports in STM32

Figure 25. Output configuration



In STM32 to make the GPIO port operational the designer has to:

- Turn on clock for the port (RCC_IOPENR)
- Set the port mode (GPIOx_MODER)
- Set the output type (GPIOx_OTYPER)
- Set speed (GPIOx_OSPEEDR)
- Set internal pull up/down resistors (GPIOx_PUPDR)

WUT Image source: ST Microelectronics website ISE

This is additional material.

External peripherals – General Purpose Input Output ports

Alternate functions

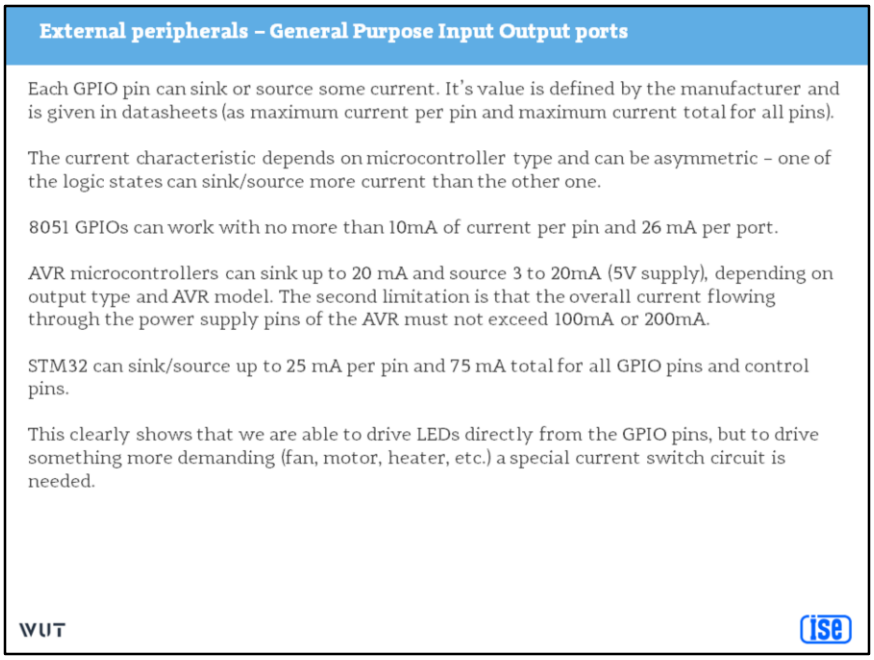
Table 14. Alternate functions selected through GPIOA_AFR registers for port A

Pin name	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7
PA0	-	USART2_CTS	TIM2_CH1_ETR	TSC_G1_IO1	-	-	-	-
PA1	EVENTOUT	USART2_RTS	TIM2_CH2	TSC_G1_IO2	-	-	-	-
PA2	-	USART2_TX	TIM2_CH3	TSC_G1_IO3	-	-	-	-
PA3	-	USART2_RX	TIM2_CH4	TSC_G1_IO4	-	-	-	-
PA4	SPI1_NSS, I2S1_WS	USART2_CK	USB_NOE	TSC_G2_IO1	TIM14_CH1	-	-	-
PA5	SPI1_SCK, I2S1_CK	CEC	TIM2_CH1_ETR	TSC_G2_IO2	-	-	-	-
PA6	SPI1_MISO, I2S1_MCK	TIM3_CH1	TIM1_BKIN	TSC_G2_IO3	-	TIM16_CH1	EVENTOUT	-
PA7	SPI1_MOSI, I2S1_SD	TIM3_CH2	TIM1_CH1N	TSC_G2_IO4	TIM14_CH1	TIM17_CH1	EVENTOUT	-
PA8	MCO	USART1_CK	TIM1_CH1	EVENTOUT	CRS_SYNC	-	-	-
PA9	-	USART1_TX	TIM1_CH2	TSC_G4_IO1	I2C1_SCL	MCO	-	-
PA10	TIM17_BKIN	USART1_RX	TIM1_CH3	TSC_G4_IO2	I2C1_SDA	-	-	-
PA11	EVENTOUT	USART1_CTS	TIM1_CH4	TSC_G4_IO3	CAN_RX	I2C1_SCL	-	-
PA12	EVENTOUT	USART1_RTS	TIM1_ETR	TSC_G4_IO4	CAN_TX	I2C1_SDA	-	-
PA13	SWDIO	IR_OUT	USB_NOE	-	-	-	-	-
PA14	SWCLK	USART2_TX	-	-	-	-	-	-
PA15	SPI1_NSS, I2S1_WS	USART2_RX	TIM2_CH1_ETR	EVENTOUT	-	USB_NOE	-	-

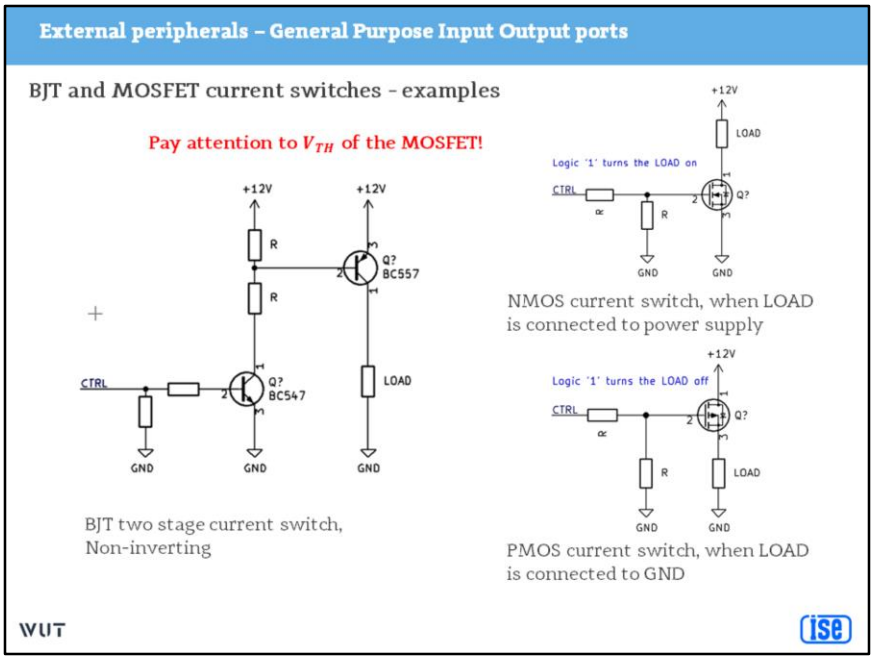
Available functions for STM32 GPIO pins (STM32F042G4U6)

WUT Image source: ST Microelectronics website ISE

This is additional material.



A very important remark – the current capability of the GPIO pin. In most cases it is limited and not equal for 0 and 1 states. It needs to be checked in the datasheet for the specific microcontroller.



These are very important circuits and for sure they will appear in Test 1.

These circuits show how to drive high current loads using a MCU GPIO pin. The left schematic shows an example of bipolar transistor circuit. The left resistor (shorting CTRL to ground) should be high resistance, like for instance 47kOhms. It is used in case the CTRL pin is floating (or high resistance state), forcing some fixed weak state to the input of lower NPN transistor. If logic one is present on CTRL pin the bottom transistor is turned on, shorting the base of the second transistor to ground (almost) and by doing so it turns on the second transistor that lets the current flow through the load.

In cases when the load does not need to be tied do ground the second transistor is not required and the load (LED and limiting resistor for example) can be connected directly to the first transistor, instead of its collector resistors.

Examples to the right show so called current switches based on MOSFET transistors. When designing such circuits you must carefully select MOSFET transistors so their VTH (Threshold Voltage).

For the NMOS transistor switch the VTH must be lower than the logic 1 voltage. Only then you are able to drive the transistor.
For the PMOS transistor switch the VTH must be high enough so that the the difference between the power supply voltage and the VTH is smaller than the logic one voltage.

Some examples. The NMOS switch is supplied (drain node) with +12V and it's VTH is 3.5V. If you try to control it with 3.3V MCU (means 3.3V logic 1) you will fail, because VTH is higher than the logic 1 voltage. If you try to drive it with 5V MCU you will succeed.
The PMOS switch is supplied (drain node) with +12V and it's VTH is 2V. This means that the difference between the power supply and VTH is 12V – 2V = 10V. Your MCU must drive this PMOS switch with logic 1 voltage higher than 10V to be able to control it. In most cases it requires additional transistor, or reducing drain power supply voltage. If we reduce the power supply voltage to 4V, then the VTH difference is 2V and we are able to control the switch using 3.3V MCU. The question is – does the load accept +4V instead of 12V?