Курс по STM32

Лекция #4:

- Периферия к GPIO: энкодер, пьезодинамик.
- Выдача альтернативы в рамках ДЗ №2.
- Архитектура и язык ассемблера ARMv6.
- Использование отладчика GDB.

25.02.2023 / 27.02.2023

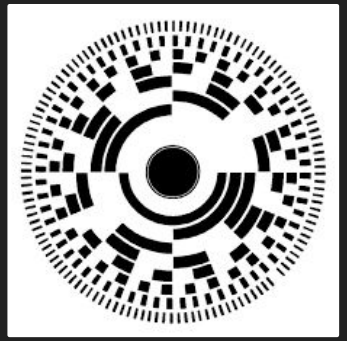
Периферия к GPIO: энкодер и пьезодинамик

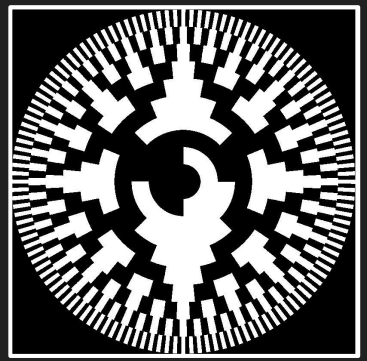
Периферия к GPIO: энкодер

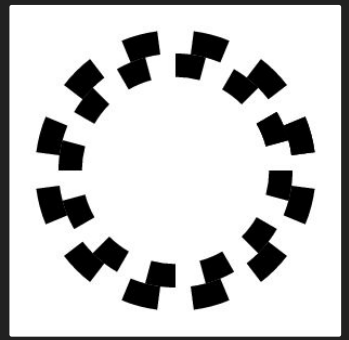
Задача энкодера – определение угла поворота.

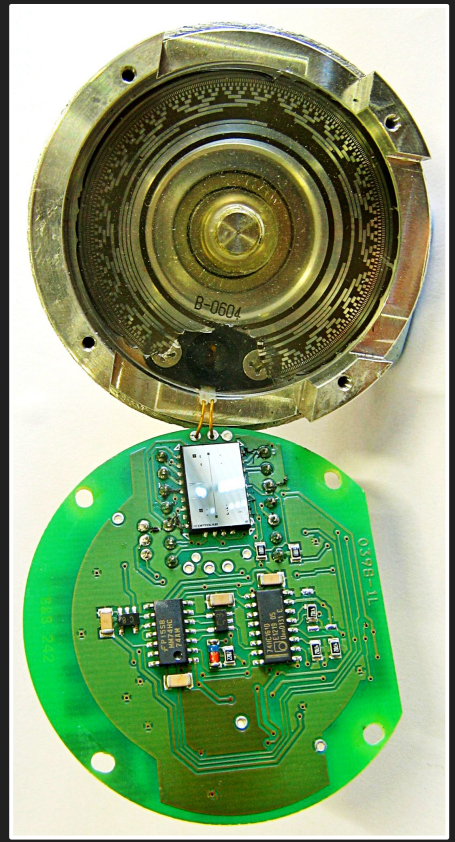
Принцип работы:
- Механический, оптический, магнитный.

Типы энкодеров:
- Абсолютный энкодер (бинарный код, код Грея)
- Инкрементальный энкодер









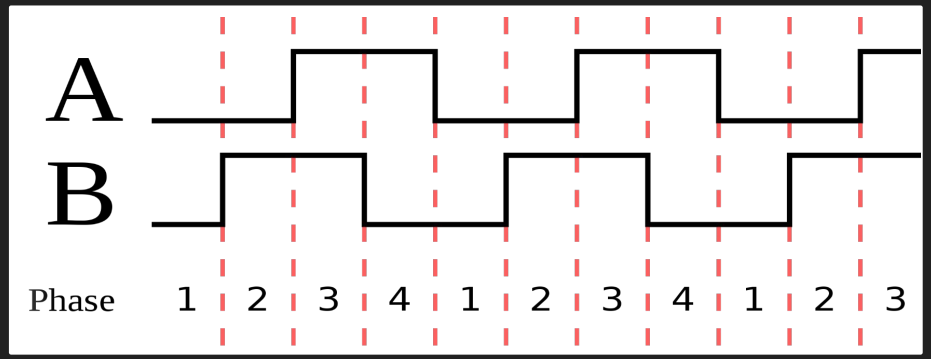
Периферия к GPIO: инкрементальный энкодер

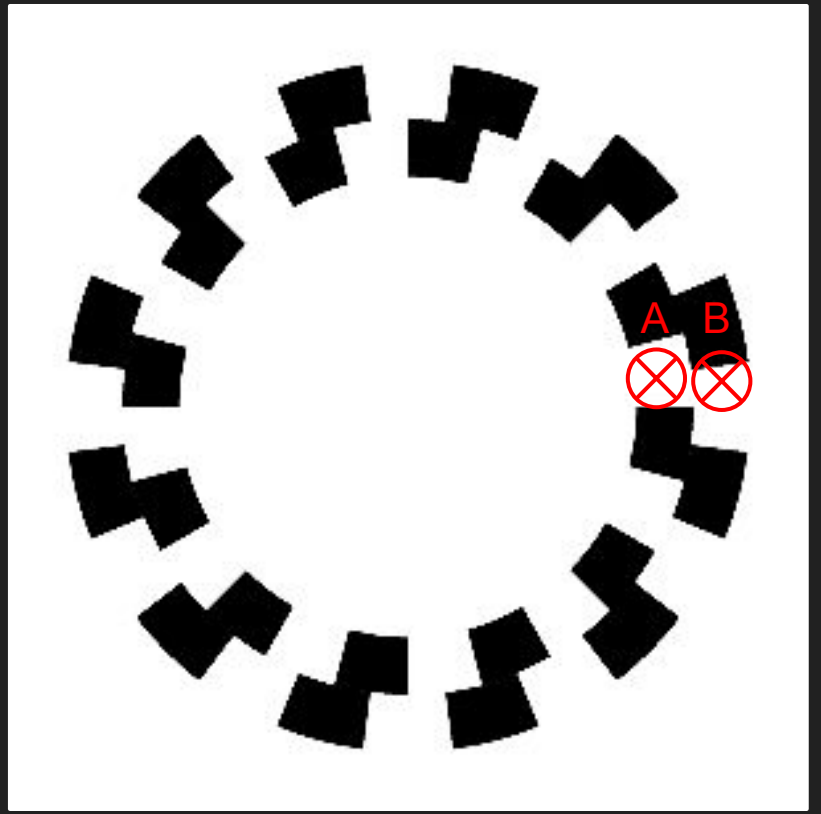
Вращение **по часовой**:

```
A 0 0 1 1 0 ...
B 0 1 1 0 0 ...
```

Вращение **против часовой**:

```
A 0 1 1 0 0 ...
B 0 0 1 1 0 ...
```

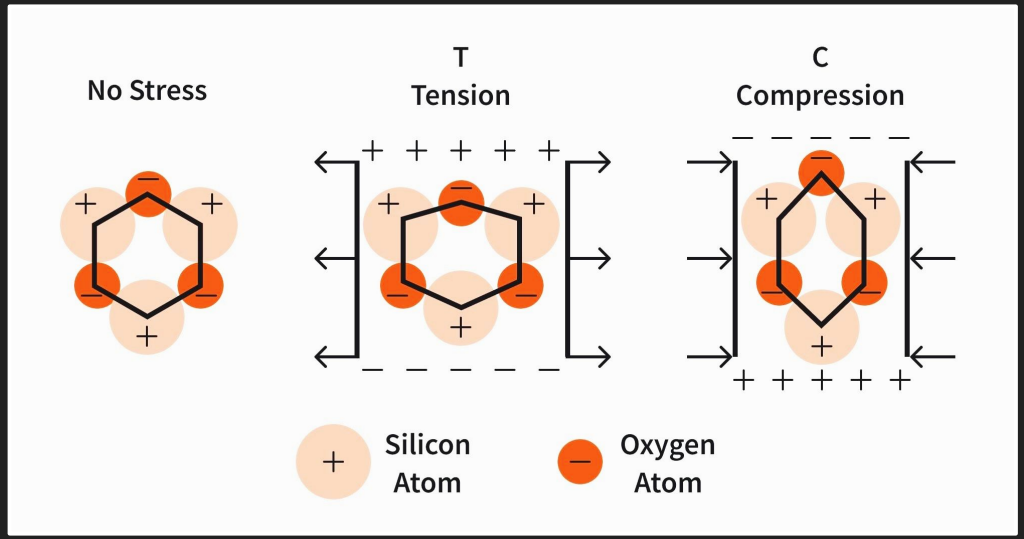


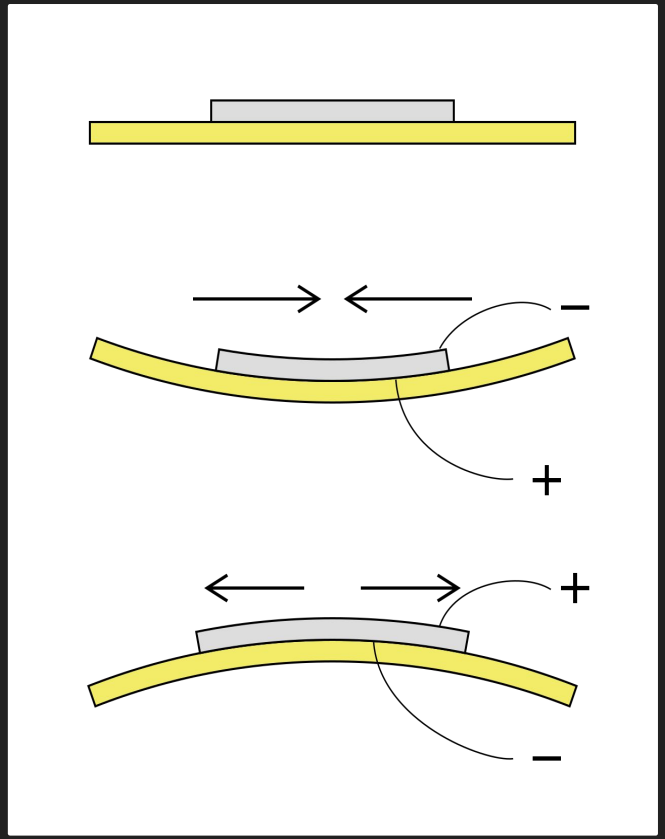


Периферия к GPIO: пьезопищалка

Пьезодинамик:

- Акустическая волна по переключению.
- Частота сигнала задаёт частоту звука.





Альтернатива в ДЗ №2

Альтернатива в рамках ДЗ №2

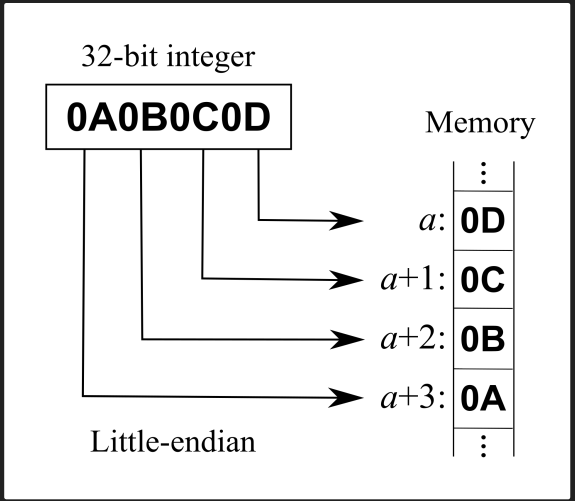
[2’] Реализовать “психоакустический излучатель”.
    [ ] Пьезодинамик звучит на заданной частоте.
    [ ] Энкодер управляет частотой звучания.
    [ ] На семисегментнике отображена частота.
    [ ] Возможно подключить несколько пьезодинамиков.
[2’] Применить “психоакустический излучатель”
    [ ] Измерить границы своего диапазона слышимости.
[3’] Реализовать схему устранения дребезга энкодера.

Архитектура ARMv6

Архитектура ARMv6: типы данных

Общие положения:

- Эндианность – `little-endian`.
- Поддерживаемые типы данных:
  `word`, `half word`, `byte`.
- Размер инструкции: 16 бит / 32 бита.
- Размер адресного пространства: 4 Гб.
- Размер шины памяти: 32 бита.




```
ldr   = Load Word                  str   = Store Word
ldrh  = Load unsigned Half Word    strh  = Store unsigned Half Word
ldrsh = Load signed Half Word      strsh = Store signed Half Word
ldrb  = Load unsigned Byte         strb  = Store unsigned Byte
ldrsb = Load signed Bytes          strsb = Store signed Byte
```

Архитектура ARMv6: режимы процессора

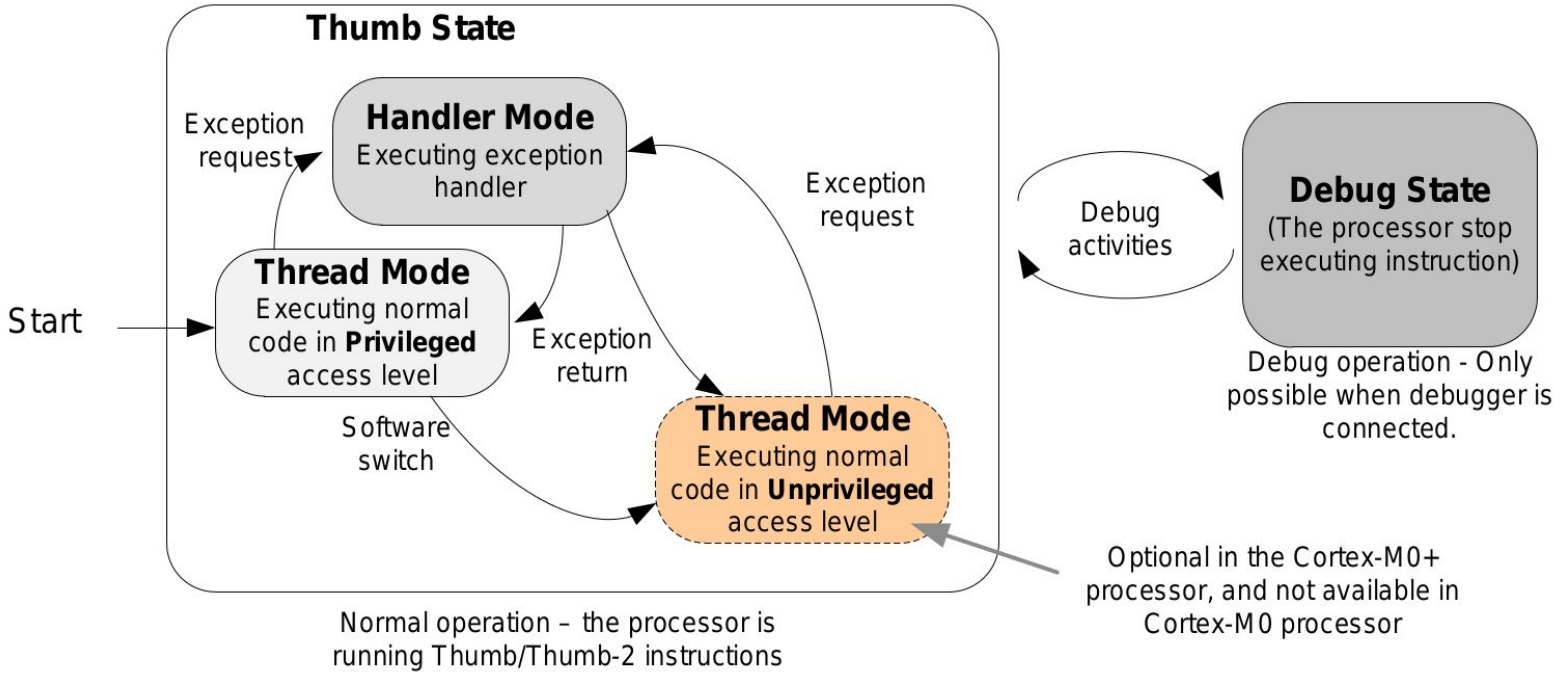
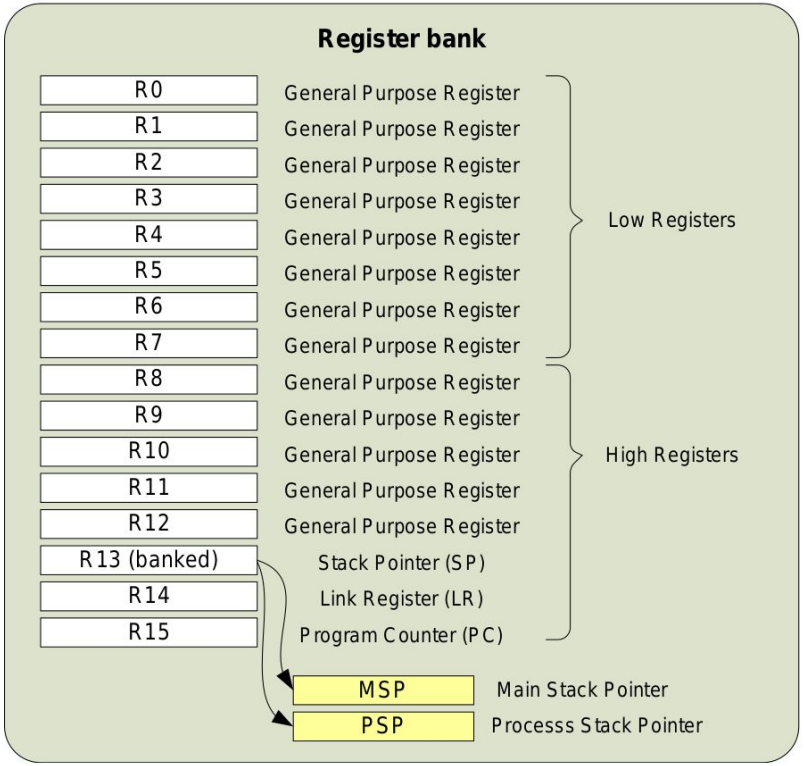


Figure 4.2
Processor modes and state in ARMv6-M architecture.

Архитектура ARMv6: регистры процессора



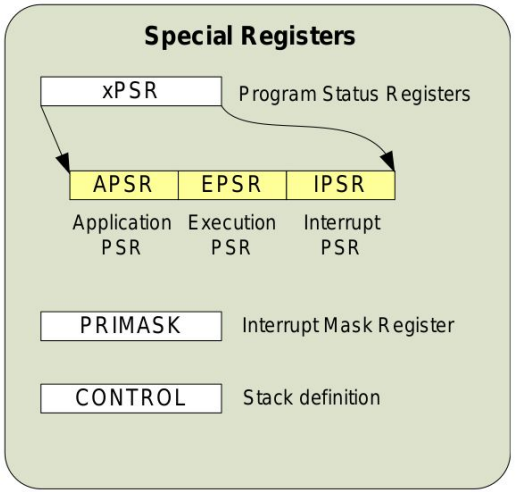


Figure 4.3
Registers in the Cortex®-M0 and Cortex-M0+ processors.

Архитектура ARMv6: специальные регистры

Флаги регистра APSR

N – Negative, Z - Zero, C - Carry, V - oVerflow, T - Thumb bit

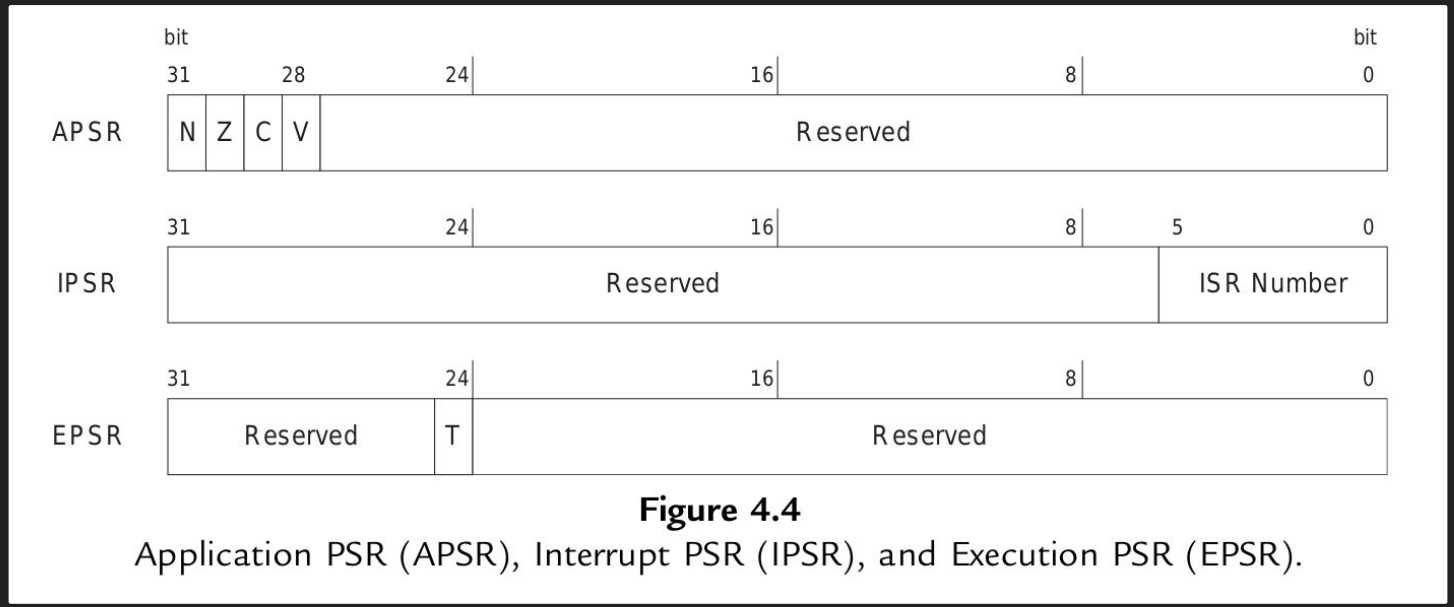


Figure 4.4
Application PSR (APSR), Interrupt PSR (IPSR), and Execution PSR (EPSR).

Архитектура ARMv6: специальные регистры

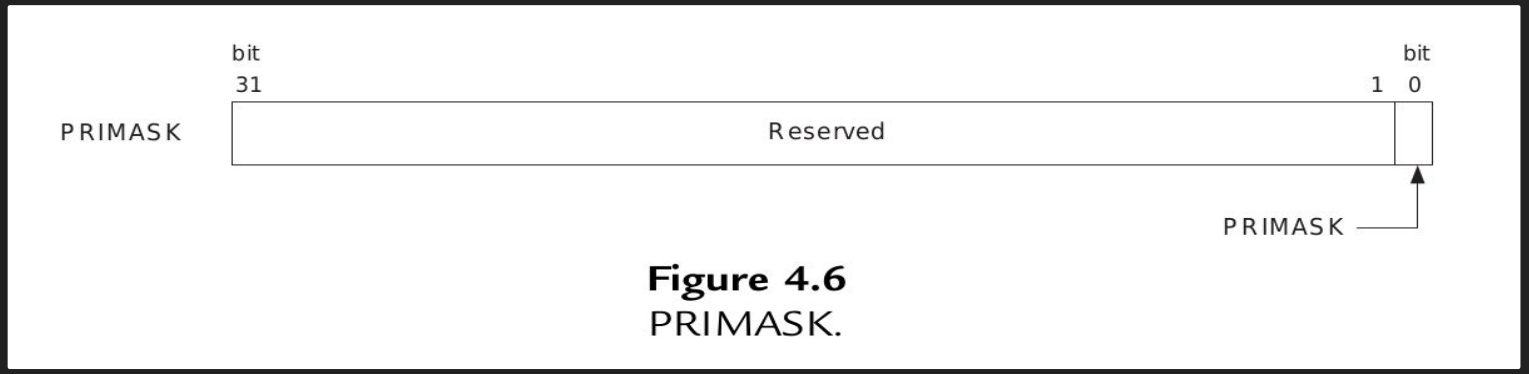


Figure 4.6
PRIMASK.

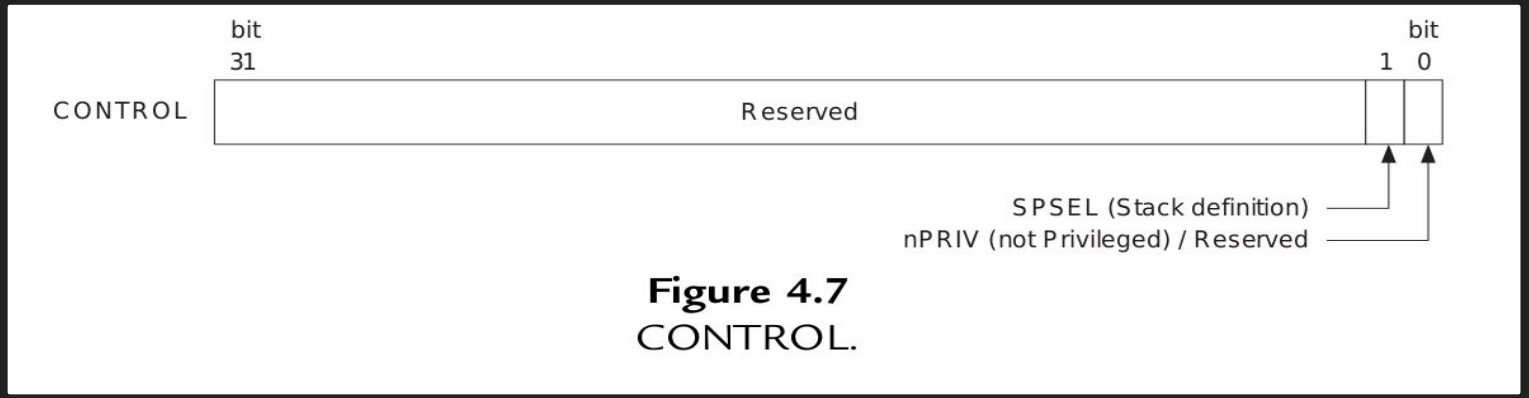


Figure 4.7
CONTROL.

Архитектура ARMv6: набор команд

Table 5.1: 16-bit Thumb® instructions supported on the Cortex®-M0 and Cortex-M0+ processor

16-bit Thumb instructions supported on Cortex-M0/M0+ processors									
ADC	ADD	ADR	AND	ASR	B	BIC	BLX	BKPT	BX
CMN	CMP	CPS	EOR	LDM	LDR	LDRH	LDRSH	LDRB	LDRSB
LSL	LSR	MOV	MVN	MUL	NOP	ORR	POP	PUSH	REV
REV16	REVSH	ROR	RSB	SBC	SEV	STM	STR	STRH	STRB
SUB	SVC	SXTB	SXTH	TST	UXTB	UXTH	WFE	WFI	YIELD

Table 5.2: 32-bit Thumb® instructions supported on the Cortex®-M0 and Cortex-M0+ processor

32-bit Thumb instructions supported on Cortex-M0/M0+ processors					
BL	DSB	DMB	ISB	MRS	MSR

Архитектура ARMv6: набор команд

```
MOV/MNV     – перемещение / перемещение + инверсия
ADD/SUB     – сложение/вычитание
MUL         – умножение
LSL/LSR/ASR – логический/арифметический сдвиг влево/вправо
CMP         – сравнение
AND/ORR/EOR – операции &, |, XOR
LDR/STR     – чтение/запись в памяти
LDM/STM     – load/store multiple
PUSH/POP    – работа со стеком
B/BL/BX/BLX – ветвление (смена состояния / обновление LR)
SVC         – системный вызов
DSB/DMB/ISB – барьеры памяти (для когерентности кэшей)
WFI/WFE     – переходы в режим низкого энергопотребления
```

Язык ассемблера ARMv6: инструкции

```
MNEMONIC{S}{condition} {Rd}, Operand1, Operand2
MNEMONIC      - Краткое наименование инструкции.
{S}           - Разрешение обновления битов регистра APSR.
{condition}   - Условие исполнения инструкции.
{Rd}          - Регистр – место назначения (destination).
Operand1      - Первый операнд (регистр или immediate).
Operand2      - Второй операнд, опционально.
                Immediate или регистр с задаваемым сдвигом.
```

```
add   r0, r1, r2      ; R0 = R1 + R2
add   r0, r1, #2      ; r0 = r1 + 2
movle r0, #5          ; if (Z == 1 and N != V) r0 = 5
mov   r0, r1, lsl #1  ; r0 = r1 << 1
```

Архитектура ARMv6: что читать?

Список справочных материалов по архитектуре и языку ассемблера ARMv6:

- Документация на процессор (docs/cortex_m0_gug.pdf, часть 3).
- Туториал по ассемблеру ARMv6.
- Книга The Definitive Guide to ARM Cortex-M0 (docs/cortex_m0_definitive_guide.pdf, части 4.1, 4.2, 5.3, 5.4, 5.5, 6)

Использование отладчика GDB

Использование отладчика: общая схема

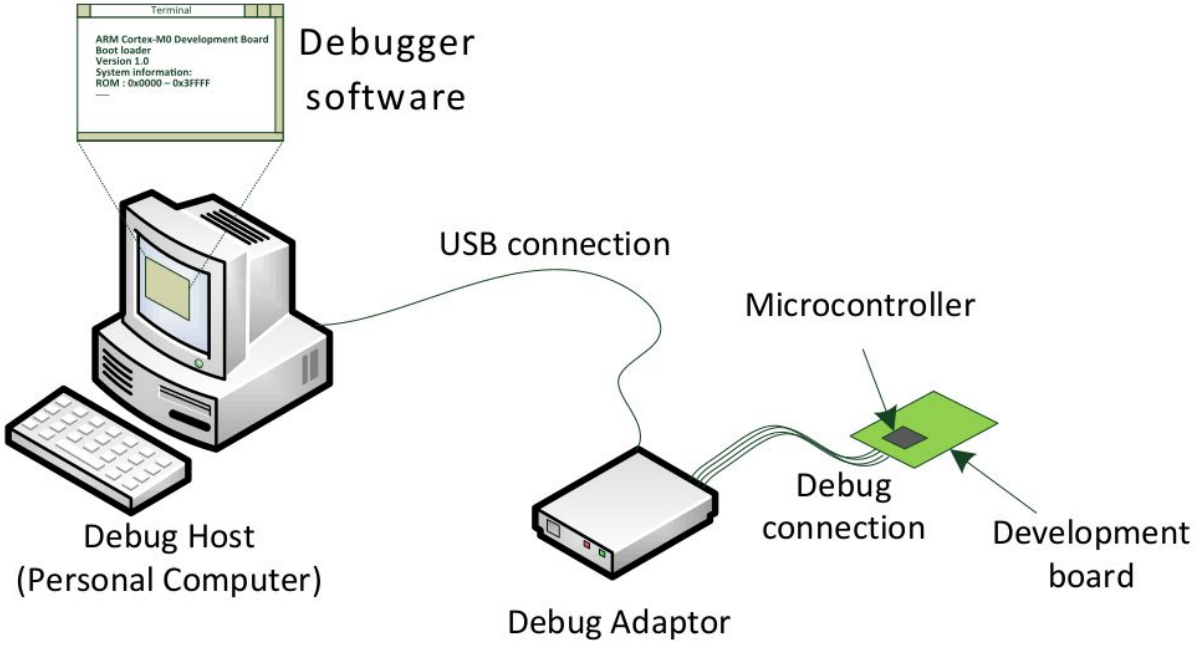


Figure 13.1

A classic microcontroller development environment.

Использование отладчика: подключение по SWD

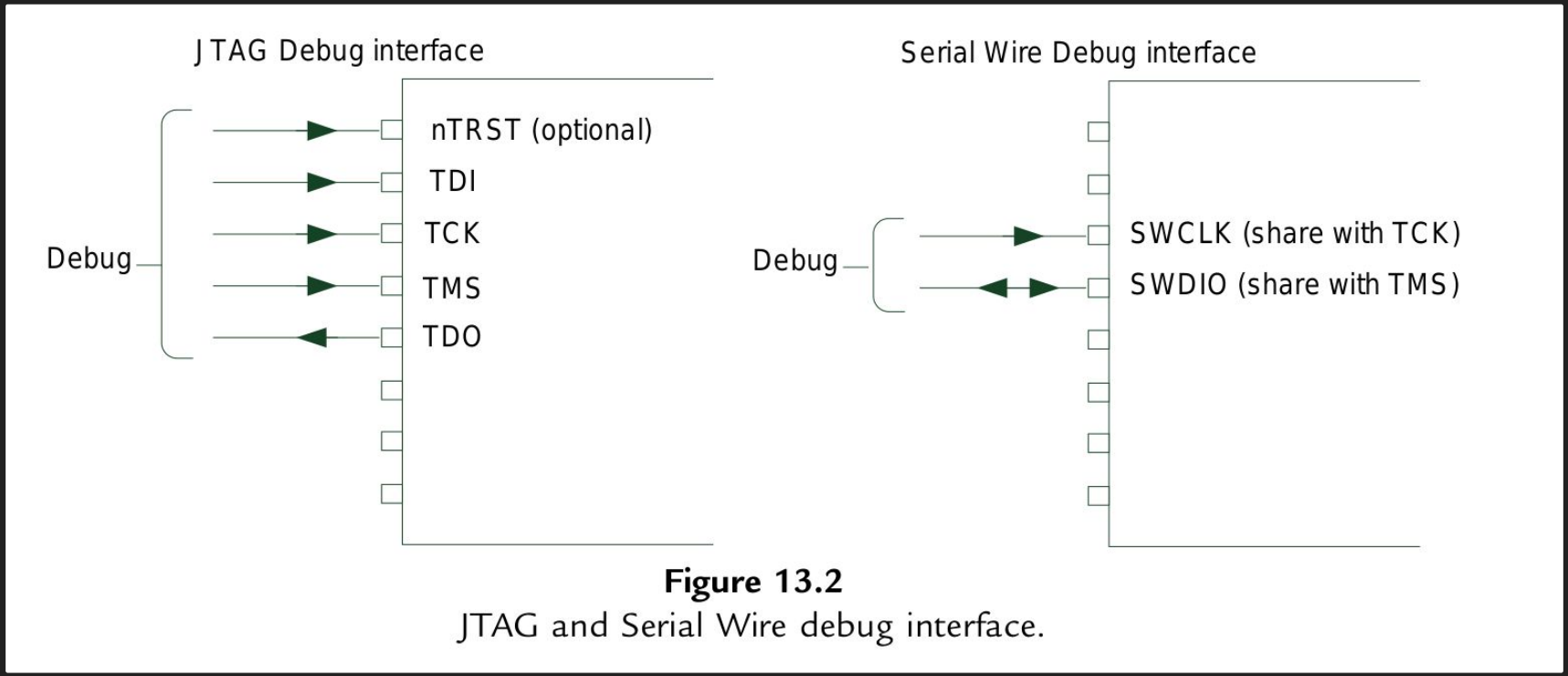
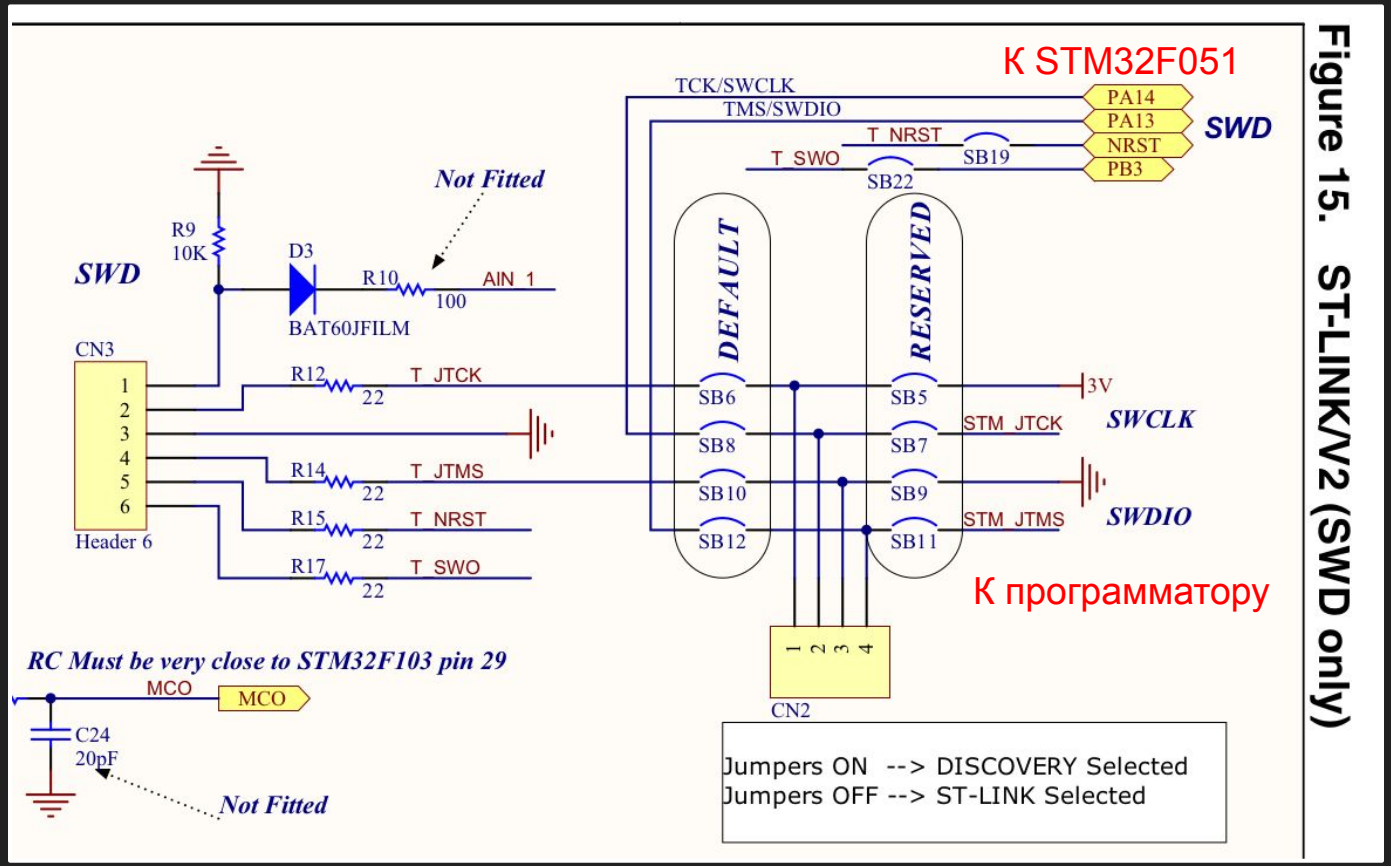


Figure 13.2
JTAG and Serial Wire debug interface.

Использование отладчика: подключение по SWD



Использование отладчика GDB: устройство

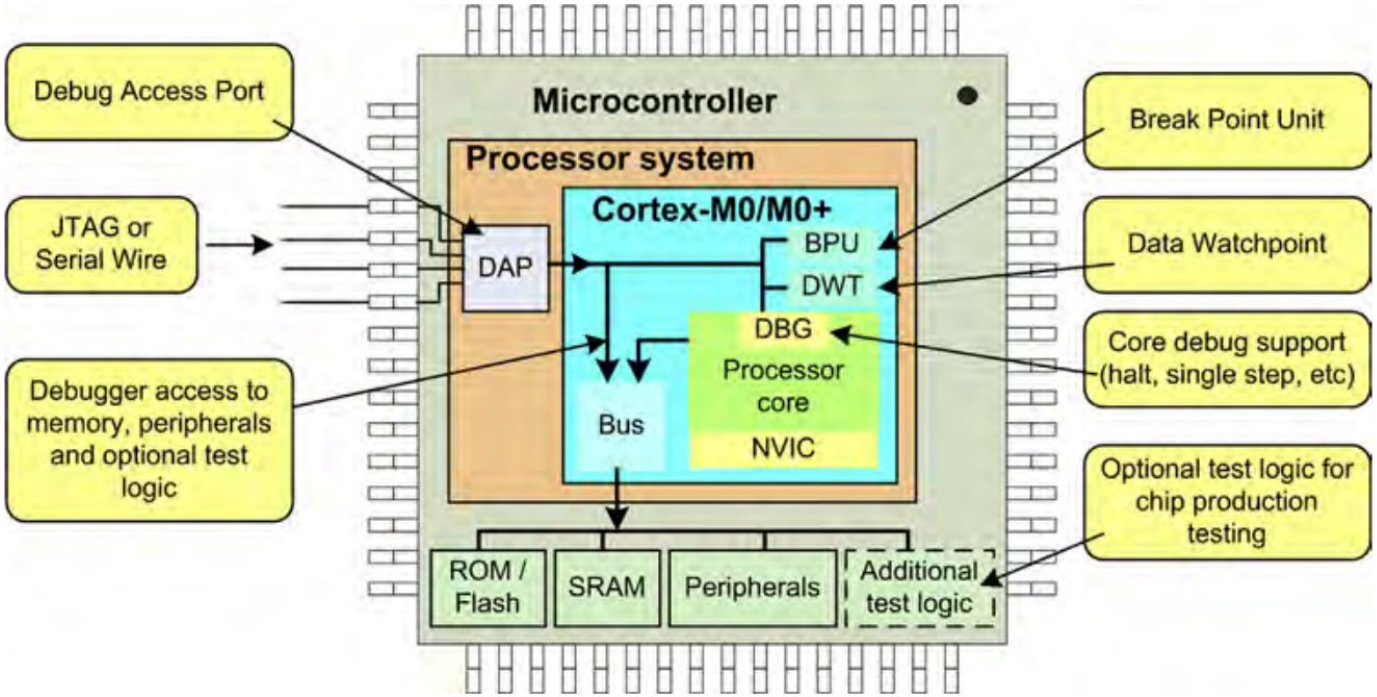


Figure 13.3
Debug interface connection inside the processor.

Использование отладчика GDB: команды

```
~/path/to/stm32f051_rewind/labs/01_blinkled>
❯ make DEBUG=1
...
❯ make flash
...
❯ make hardware
st-util -p 1234
st-util 1.6.0
2023-02-25T17:53:29 INFO common.c: Loading device parameters....
2023-02-25T17:53:29 INFO common.c: Device connected is: F0 device, id
0x20006440
2023-02-25T17:53:29 INFO common.c: SRAM size: 0x2000 bytes (8 KiB),
Flash: 0x10000 bytes (64 KiB) in pages of 1024 bytes
2023-02-25T17:53:29 INFO gdb-server.c: Listening at *:1234...
```

Использование отладчика GDB: команды

```
~/path/to/stm32f051_rewind/labs/01_blinkled>
❯ make gdb
...
(gdb) tui e
<Рендер красивого окошка с кодом на ассемблере/си>
(gdb) la src / la asm / la regs
<Рендер окошка с кодом на си, на ASM, с регистрами>
(gdb) foc n / foc p
<Переход между окошками>
(gdb) s / si / n / ni
<s/n - шаг вглубь / шаг через; _/i - строка/инструкция>
(gdb) b <имя функции>
<Установка breakpoint-а>
(gdb) c
<Продолжение выполнения программы>
```

Спасибо за внимание!

Красивые иконки взяты с сайта handdrawngoods.com