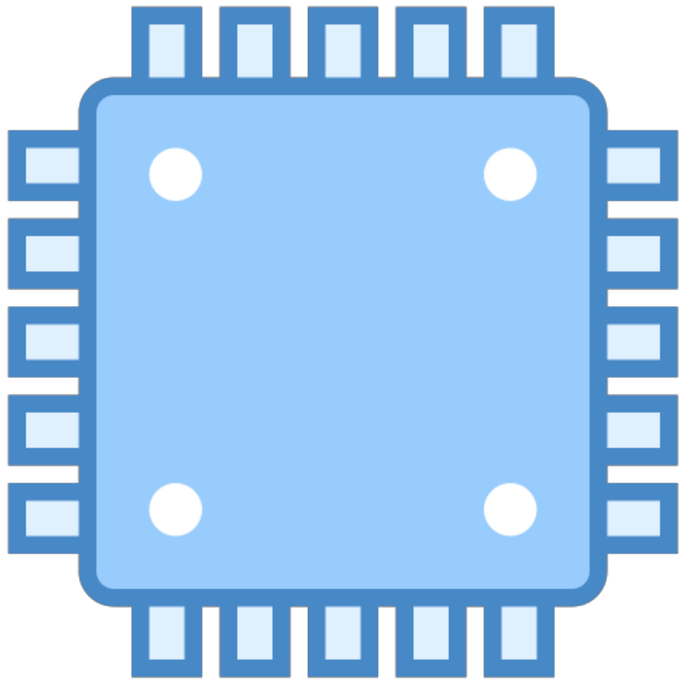


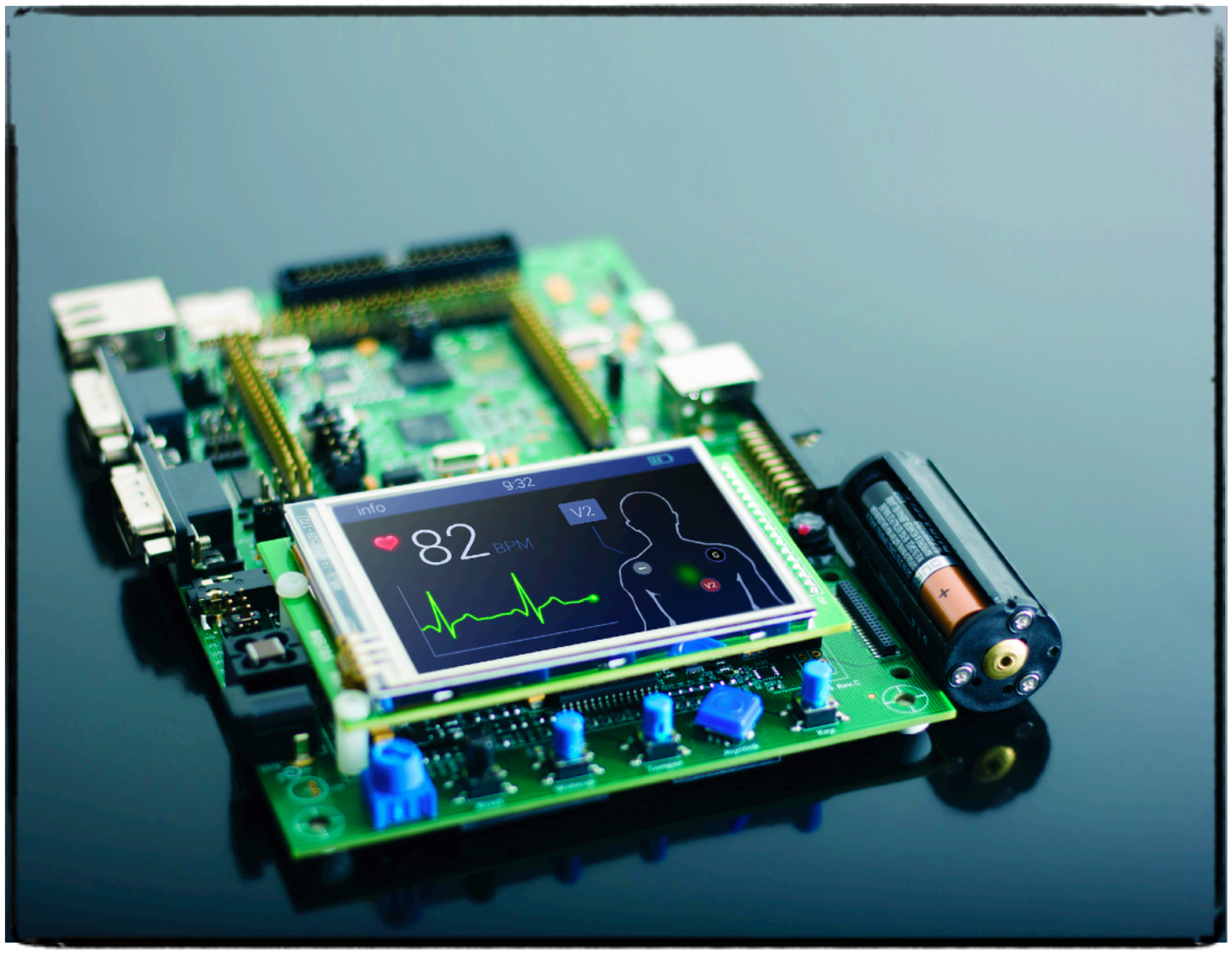
Introduction to AVR Assembly

by Mehdi Safaee

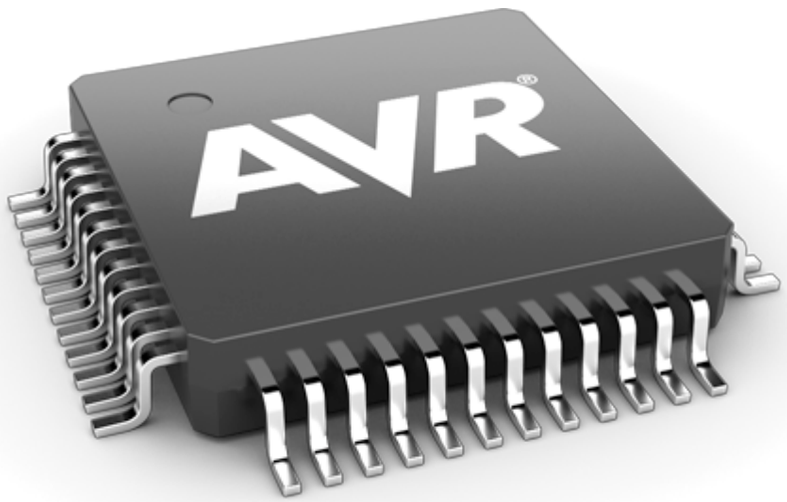
Subjects

1. Introduction
2. AVR assembly useful instructions
3. A practical example

Embedded systems

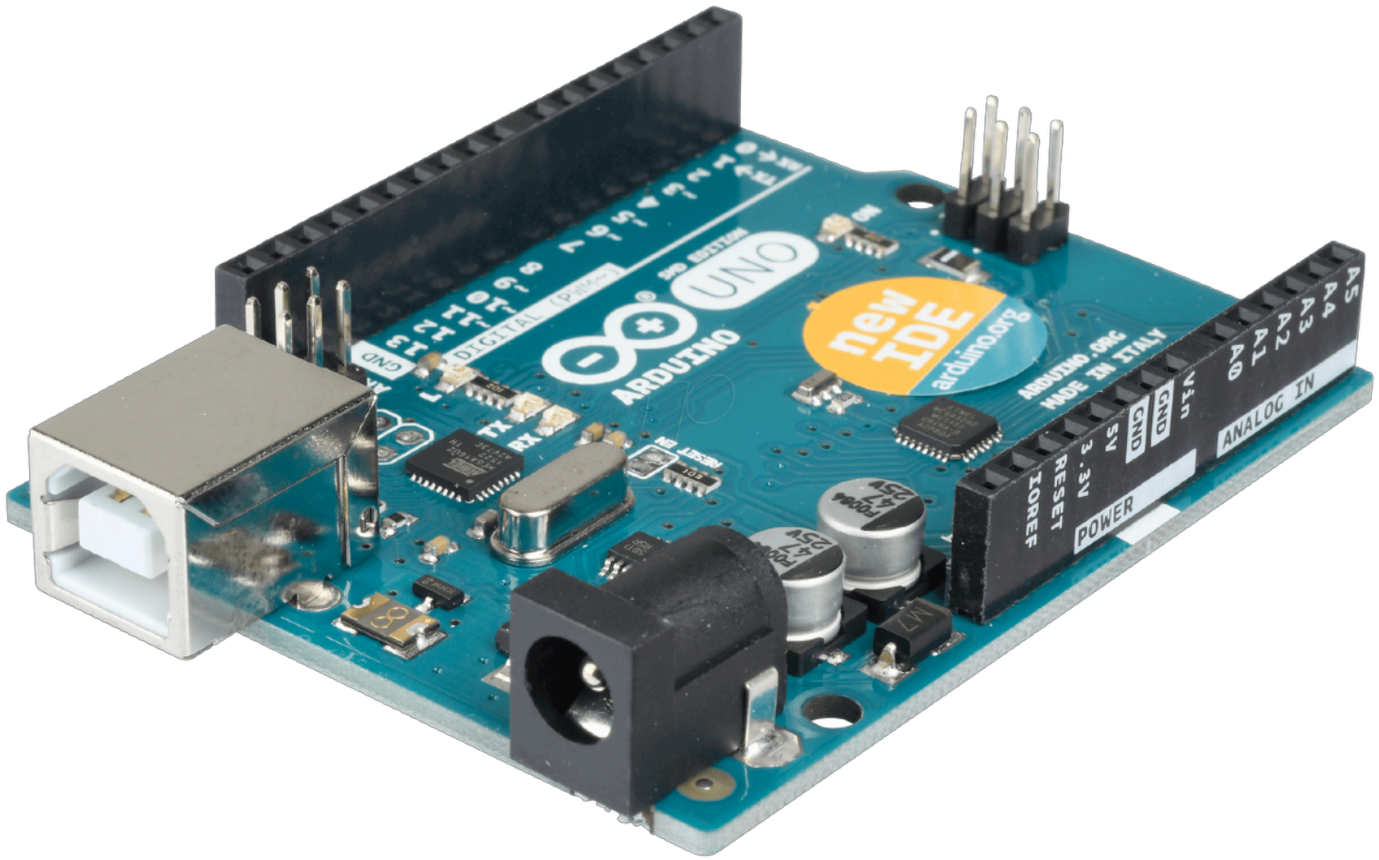


Famous architectures



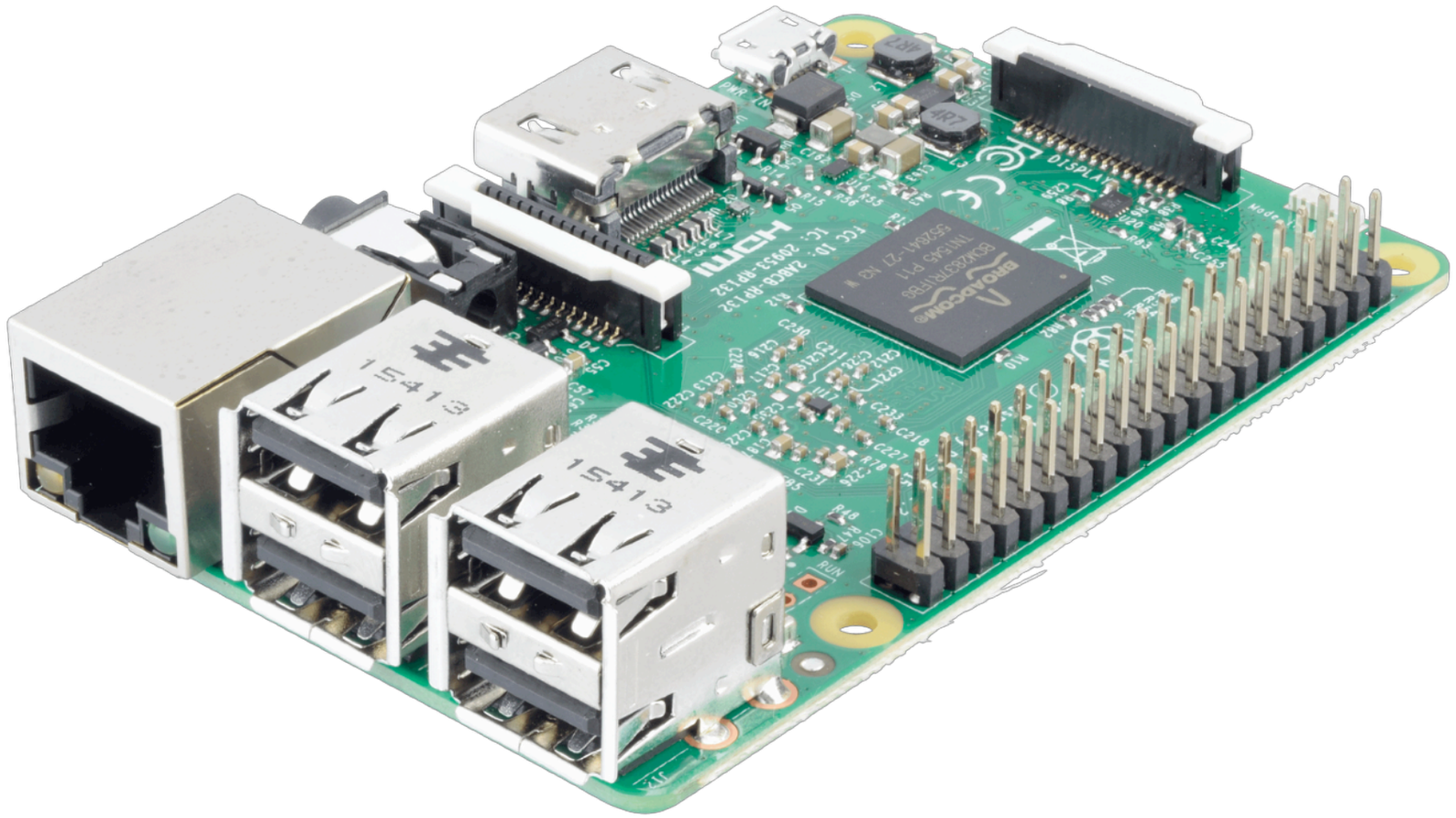
Arduino

Note 1 Based on ATmega328



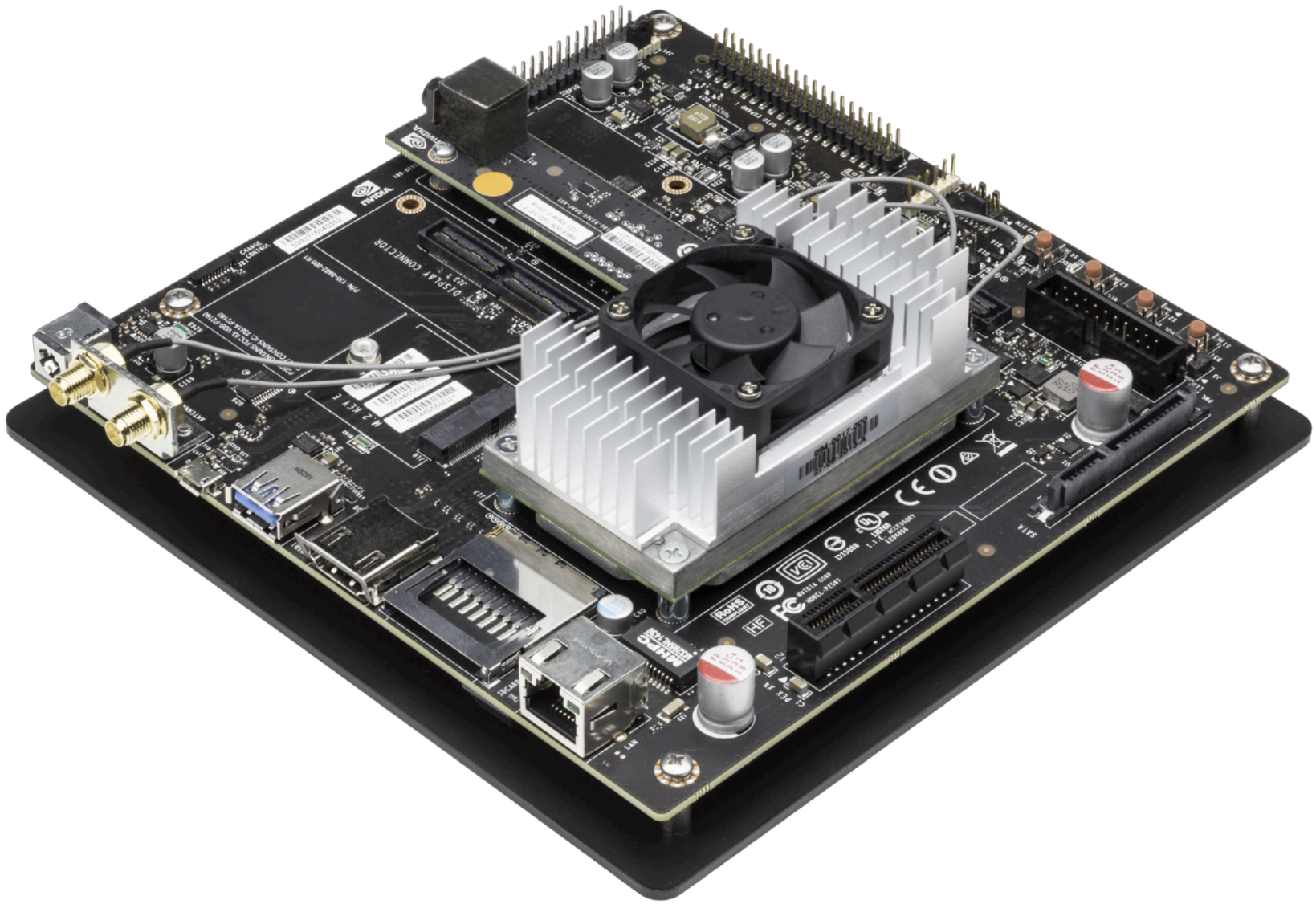
Raspberry pi

Note 2 Based on ARM Cortex A53

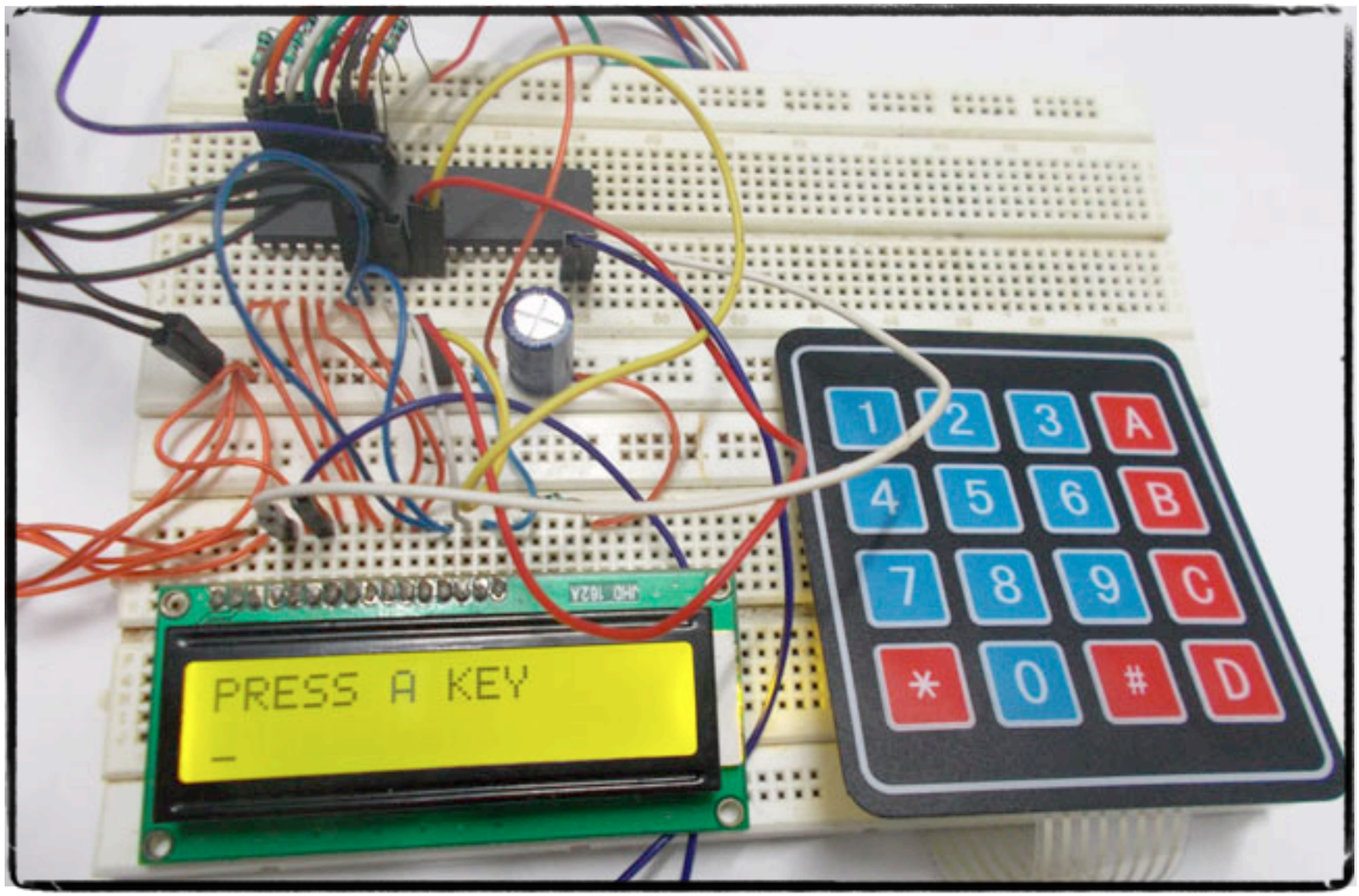


Advanced Boards

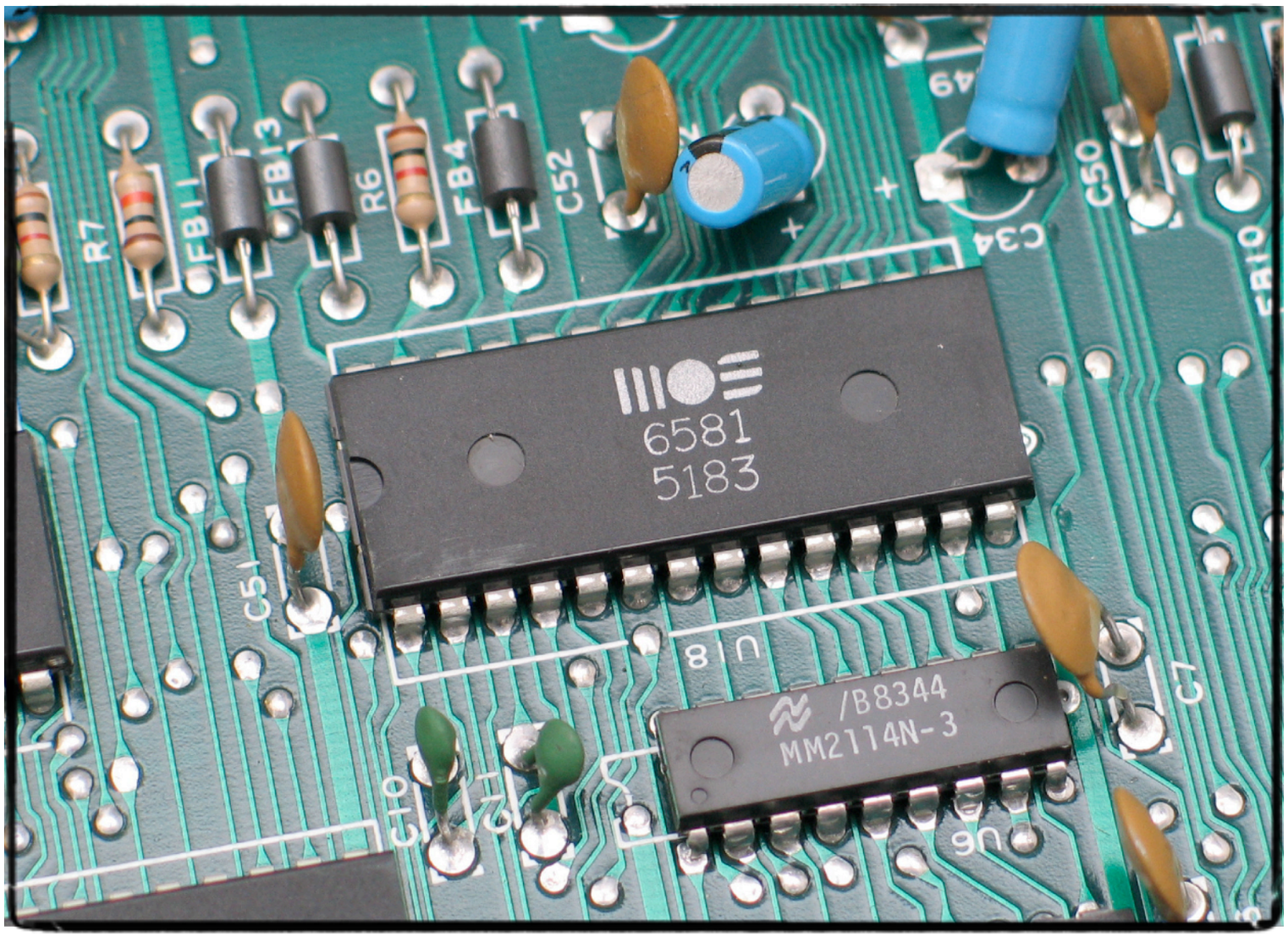
Note 3 Based on ARM Cortex A + Nvidia Pascal GPU



Baremetal embedded design



Baremetal embedded design



Subjects

1. Introduction
2. AVR assembly useful instructions
3. A practical example

.def/.include

Code

```
.def temp = r16

.include "test.inc"
```

MOV/LDI/NOP

Code

```
mov r16,r0 ; Copy r0 to r16

ldi r30, 0x20 ; load 0x20 in r30 register

nop ; Wait (do nothing)
```

Adding/Subtraction

Code

```
add r1,r2 ; Add r2 to r1 (r1=r1+r2)
add r28, r28 ; Add r28 to itself (r28=r28+r28)

sub r13,r12 ; Subtract r12 from r13
subi r22,0x14; Subtract 20 from r22

add r2,r0 ; Add low byte
adc r3,r1 ; Add with carry high byte
```

And/Or

Code

```
and r2,r3 ; Bitwise and r2 and r3, result in r2
andi r18,0x10 ; Isolate bit 4 in r18

or r15,r16 ; Do bitwise or between registers
ori r17,1 ; Set bit 0 of r17
```

Shifting

Code

```
add r0, r4 ; Add r4 to r0
lsl r0 ; Multiply r0 by 2

add r0,r4 ; Add r4 to r0
lsr r0 ; Divide r0 by 2
```

Call/JMP/ret

Code

```
; other code
call myLabel ; Call subroutine
; other code

myLabel :
    ; subroutine code
ret


jmp myLabel ; Jump to a subroutine
continueLabel :

myLabel :
    ; subroutine code
    ; ...
jmp continueLabel
```

Comparing and branching

Code

```
cpi r19,3 ; Compare r19 with 3
brne myLabel ; Branch if r19 != 3

cp r4,r19 ; Compare r4 with r19
brne myLabel ; Branch if r4 != r19

cp r11,r12 ; Compare registers r11 and r12
brge myLabel ; Branch if r11 ≥ r12 (signed)

cp r1, r0 ; Compare registers r1 and r0
breq myLabel ; Branch if registers equal
```

Inc/Dec

Code

```
inc r22 ; increment r22

dec r17 ; Decrement r17
```

IN/OUT

Code

```
in r25, pinB ; Read Port B

ldi r25, 0x04
out portB , r25
```

SBI/CBI

Code

```
sbi portA ,1 ; Set 2nd bit in portA

cbi portA ,5 ; Clear 6th bit in portA
```

SBR/CBR/CLR

Code

```
sbr r16,3 ; Set 4th bit in r16

cbr r18, 1 ; Clear 2nd bit in r18

clr r18 ; Clear r18
```

SBRS/SBRC

Code

```
sbrs r0,7 ; Skip if 8th bit in r0 set
neg r0 ; Only executed if 8th bit in r0 not set

sbrc r0,7 ; Skip if bit 8th in r0 cleared
sub r0,r1 ; Only executed if 8th bit in r0 not cleared
```

SBIS/SBIC

Code

```
sbis portB,7 ; Skip if 8th bit in portB set
sub r0,r1 ; Only executed if 8th bit in r0 not cleared

sbic portB,7 ; Skip if 8th bit in portB cleared
neg r0 ; Only executed if 8th bit in r0 not set
```

NEG/COM

Code

```
com r4 ; Take one's complement of r4
neg r11 ; Take two's complement of r11
```

CLI/SEI/RETI

Code

```
sei ; set global interrupt enable
cli ; Disable interrupts
reti ; return from interrupts
```

General Pattern for Writing Assembly Programs (part 1)

template.asm

```
INCLUDES & INITIALIZATIONS

INTERRUPT VECTORS

INTERRUPT ROUTINES

MAIN LOOP

SUBROUTINES
```

Subjects

1. Introduction
2. AVR assembly useful instructions
3. A practical example

Example

Example 1 Design a simple 3x8 decoder using portA and portB of microcontroller.