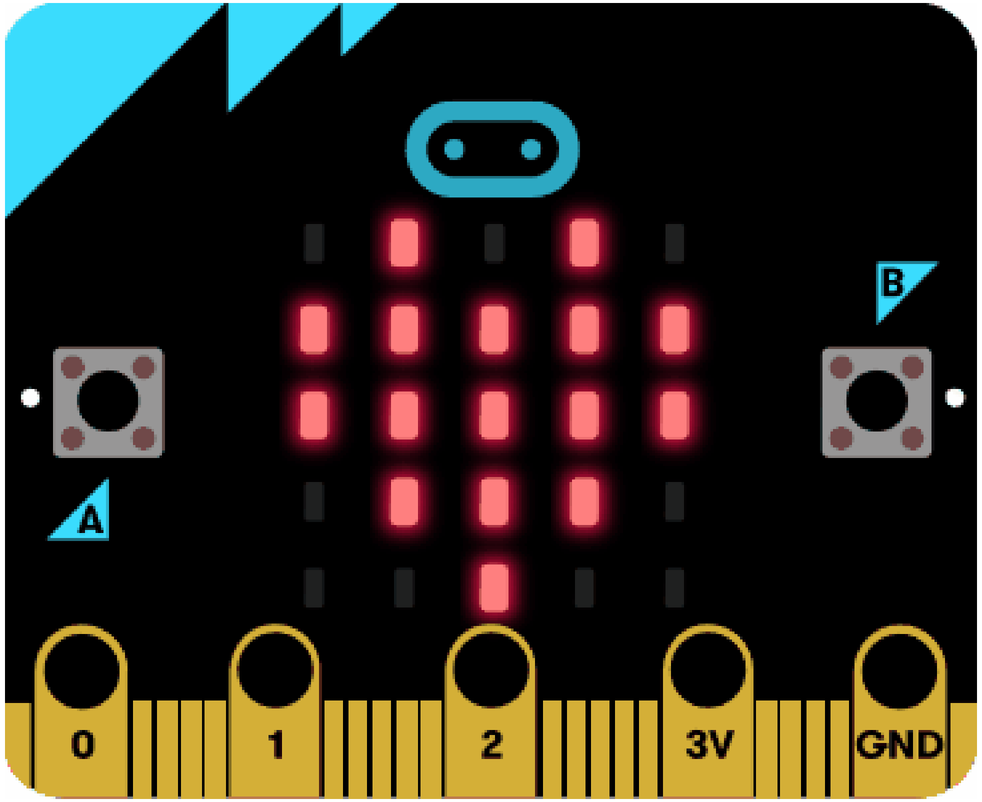
Topic 11 – LED Operation

Designing Real Systems

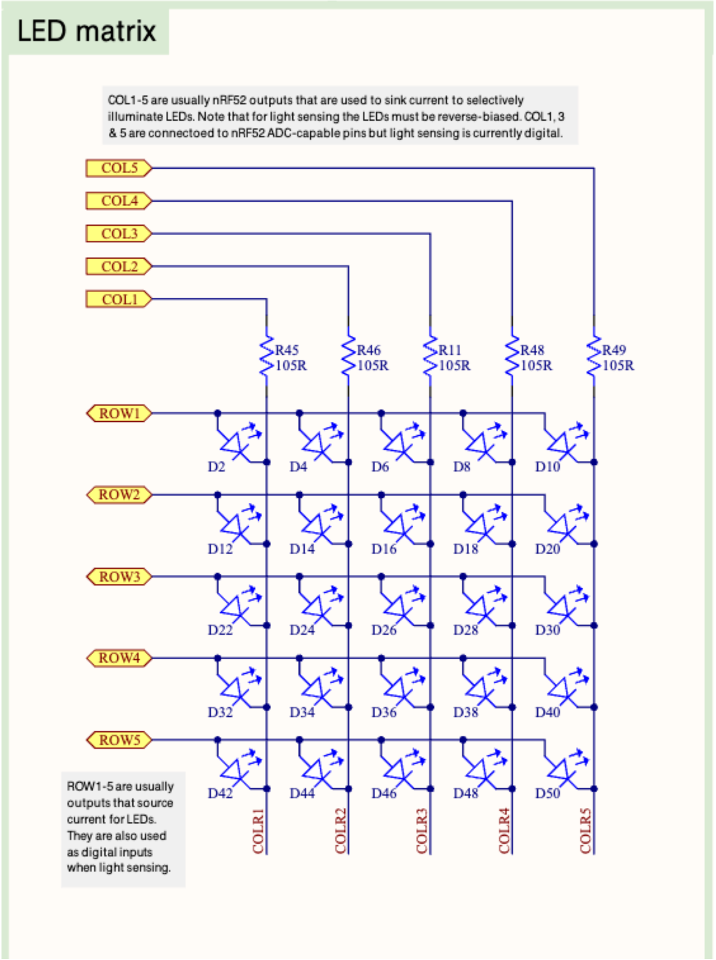
- When designing systems, several factors may complicate
  the programming model:
  - Cost, hardware limitation, energy, scalability.
- Abstractions can simplify coding, by trading CPU cycles.
- Example: operating micro:bit LEDs



LED schematic

- The **5×5 LED matrix** is wired in a way that:
  - **Columns (x-axis) are controlled separately**.
  - **Rows (y-axis) are controlled separately**.
- *Each LED is at the **intersection of a row and a column**.*
- Complex displays must draw row by row.
  - **Rows (anodes, positive side), Columns (cathodes, negative side)**

- To light up a particular LED:
  1. The corresponding **row is set HIGH** (enabled).
  2. The corresponding **column is set LOW** (to complete the circuit and turn on the LED).

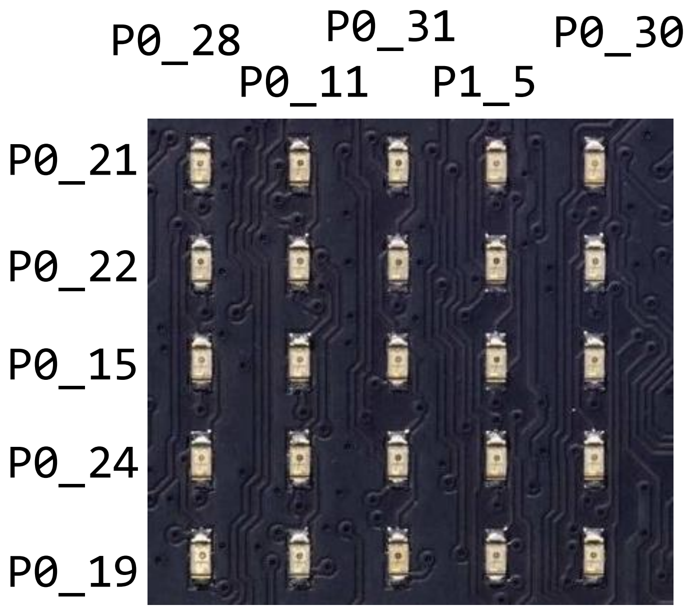


https://github.com/microbit-foundation/microbit-v2-hardware/blob/main/V2.00/MicroBit_V2.0.0_S_schematic.PDF

Micro:bit LED MMIO

- The CPU on the micro:bit uses the GPIO mechanism to control LEDs.
- There are two GPIO ports (0 and 1), with 32 I/O pins each.
- COLs and ROWs spread on both GPIO pins.

- *You need some clever programming to light complex LED patterns.*



GPIO MMIO Register

Name	Address	Description
OUT	0x50000504 - 0x50000804	Write GPIO port
IN	0x50000510 - 0x50000810	Read GPIO port
DIRSET	0x50000518 - 0x50000810	Direction of GPIO pins

DIRSET

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	f	e	d	c	b	a	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A
Reset 0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Id	RW	Field	Value Id	Value	Description
A	RW	PIN0			Set as output pin 0
			Input	0	Read: pin set as input
			Output	1	Read: pin set as output
			Set	1	Write: writing a '1' sets pin to output; writing a '0' has no effect
B	RW	PIN1			Set as output pin 1
			Input	0	Read: pin set as input
			Output	1	Read: pin set as output
			Set	1	Write: writing a '1' sets pin to output; writing a '0' has no effect
C	RW	PIN2			Set as output pin 2
			Input	0	Read: pin set as input
			Output	1	Read: pin set as output
			Set	1	Write: writing a '1' sets pin to output; writing a '0' has no effect

https://infocenter.nordicsemi.com/index.jsp?topic=%2Fcom.nordic.infocenter.nrf52832.ps.v1.1%2Fgpio.html

CODAL Display API

- The CODAL LED API is built on top of **multiplexing**, where the **5×5 LED matrix is refreshed row by row**.

- The API allows users to:
  - **Set individual LEDs** (`display.image.setPixelValue(x, y, value)`)
  - **Show pre-defined or custom images** (`display.print()`)
  - **Control brightness** (display.image.setPixelValue() with brightness level)
  - **Scroll text** (`display.scroll()`)

- Things are much more complex under the hood…

LED programming

```
@ Init LEDs
ldr r0, =0x50000514 @ GPIO0_DIR register
ldr r1, =0xd1688800 @ LED pins as outputs
str r1, [r0]
ldr r0, =0x50000814 @ Same for GPIO1_DIR
ldr r1, =0x00000020
str r1, [r0]

ldr r4, =0x50000504 @ Address of GPIO0_OUT
register
ldr r5, =0x50000804 @ Address of GPIO1_OUT
register
ldr r6, =0x50008800 @ Bit pattern 0 for dot
ldr r7, =0x00000020 @ Bit pattern 1 for dot
str r6, [r4]        @ Light an LED
str r7, [r5]
```

Set ports as outputs for every PIN used to control LEDs.

Set Row 3 and Columns 1,2,4,5 to 1.

Note: In order to light a led, you need to turn on ts input, and turn off the output, to allow current to flow.

ARM Constants

- ARM Assembly offers a set of directives to define constant values.
  - .equ: Assign a name to a constant value - immutable.

```
.set GPIO1_OUT, 0x50000804
```

  - .set: Assign a name to a constant value – mutable.

```
.equ ROW_1, 21
```

  - Look like `#define` in C.
  - Assembler will pattern match replace during binary conversion.

```
ldr r5, =GPIO1_OUT
```

LED initialization function

- Need to set the GPIO LED pins as output pins.

```
led_init:
 ldr r0, =GPIO0_DIR @ GPIO0_DIR register
 ldr r1, =0xd1688800 @ LED pins as outputs
 str r1, [r0]
 ldr r0, =GPIO1_DIR @ Same for GPIO1_DIR
 ldr r1, =0x00000020
 str r1, [r0]
 bx lr
```

Turn on a single LED

- In order to turn on LED (1, 1) need to turn on row 1 and columns 2,3,4,5.

```
mov r1, #0x1
mov r6, #0x0
orr r6, r6, r1, lsl ROW_1
orr r6, r6, r1, lsl COL_2
orr r6, r6, r1, lsl COL_3
orr r6, r6, r1, lsl COL_5
str r6, [r4] @ Light an LED
mov r6, #0x0 @ P1.5
orr r6, r6, r1, lsl COL_4
str r6, [r5] @ Light an LED
```

LED multiplexing

- We cannot light up multiple rows at the same time.
  - How can we display a heart?

- Ideas: turn-on LED, line by line.
- Human eye cannot perceive the difference, if frequency is above 60Hz (i.e. 40 msec).

(0,0) **(1,0)** (2,0) **(3,0)** (4,0)
(0,1) (1,1) (2,1) (3,1) (4,1)
(0,2) (1,2) (2,2) (3,2) (4,2)
(0,3) (1,3) (2,3) (3,3) (4,3)
(0,4) (1,4) (2,4) (3,4) (4,4)

LED multiplexing

- We cannot light up multiple rows at the same time.
  - How can we display a heart?

- Ideas: turn-on LED, line by line.
- Human eye cannot perceive the difference, if frequency is above 60Hz (i.e. 40 msec).

(0,0)	(1,0)	(2,0)	(3,0)	(4,0)
(0,1)	**(1,1)**	**(2,1)**	**(3,1)**	**(4,1)**
(0,2)	(1,2)	(2,2)	(3,2)	(4,2)
(0,3)	(1,3)	(2,3)	(3,3)	(4,3)
(0,4)	(1,4)	(2,4)	(3,4)	(4,4)

LED multiplexing

- We cannot light up multiple rows at the same time.
  - How can we display a heart?
- Ideas: turn-on LED, line by line.
- Human eye cannot perceive the difference, if frequency is above 60Hz (i.e. 40 msec).

(0,0)	(1,0)	(2,0)	(3,0)	(4,0)
(0,1)	(1,1)	(2,1)	(3,1)	(4,1)
(0,2)	**(1,2)**	**(2,2)**	**(3,2)**	(4,2)
(0,3)	(1,3)	(2,3)	(3,3)	(4,3)
(0,4)	(1,4)	(2,4)	(3,4)	(4,4)

LED multiplexing

Lancaster University

- We cannot light up multiple rows at the same time.
  - How can we display a heart?
- Ideas: turn-on LED, line by line.
- Human eye cannot perceive the difference, if frequency is above 60Hz (i.e. 40 msec).

(0,0)	(1,0)	(2,0)	(3,0)	(4,0)
(0,1)	(1,1)	(2,1)	(3,1)	(4,1)
(0,2)	(1,2)	(2,2)	(3,2)	(4,2)
(0,3)	**(1,3)**	**(2,3)**	**(3,3)**	(4,3)
(0,4)	(1,4)	(2,4)	(3,4)	(4,4)

LED multiplexing

Lancaster University

- We cannot light up multiple rows at the same time.
  - How can we display a heart?
- Ideas: turn-on LED, line by line.
- Human eye cannot perceive the difference, if frequency is above 60Hz (i.e. 40 msec).

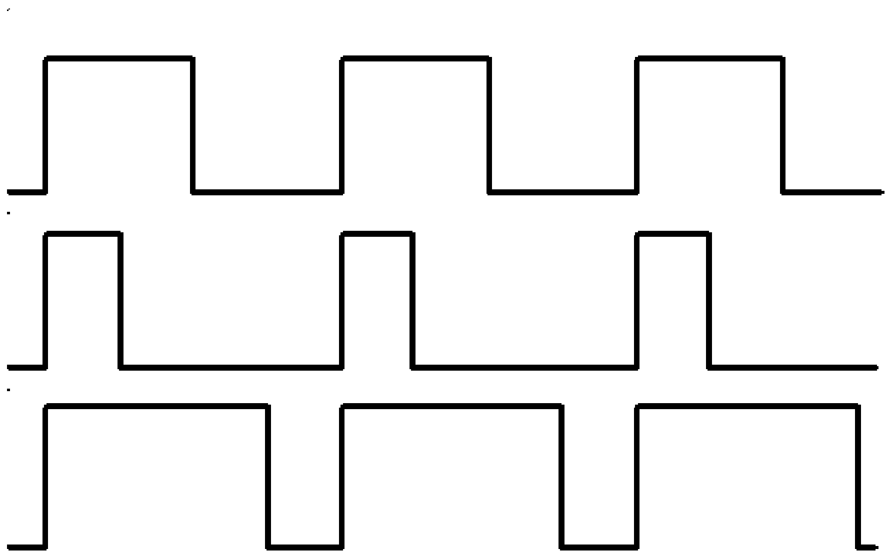
(0,0)	(1,0)	(2,0)	(3,0)	(4,0)
(0,1)	(1,1)	(2,1)	(3,1)	(4,1)
(0,2)	(1,2)	(2,2)	(3,2)	(4,2)
(0,3)	(1,3)	(2,3)	(3,3)	(4,3)
(0,4)	(1,4)	**(2,4)**	(3,4)	(4,4)

Code details

- Check sample code at: **https://scc-source.lancs.ac.uk/scc.Y1/scc.131/arm-assembly-examples/-/tree/main/lec10**

Pulse Width modulation

- The pins of your board cannot output an analog signal the way an audio amplifier can – by modulating the voltage on the pin.
- Those pins can only either enable the full 3.3V output, or pull it down to 0V.
- How does LED brightness work?
  - Switch that voltage on and off very fast.
  - Control how long it is on and how long it is off.
- This technique is called **Pulse-Width Modulation (PWM)**.



Recap

- Example LED use-case
  - LED Multiplexing concept.
  - Design code to display a heart
  - PWM concept.

- Next: lab solutions