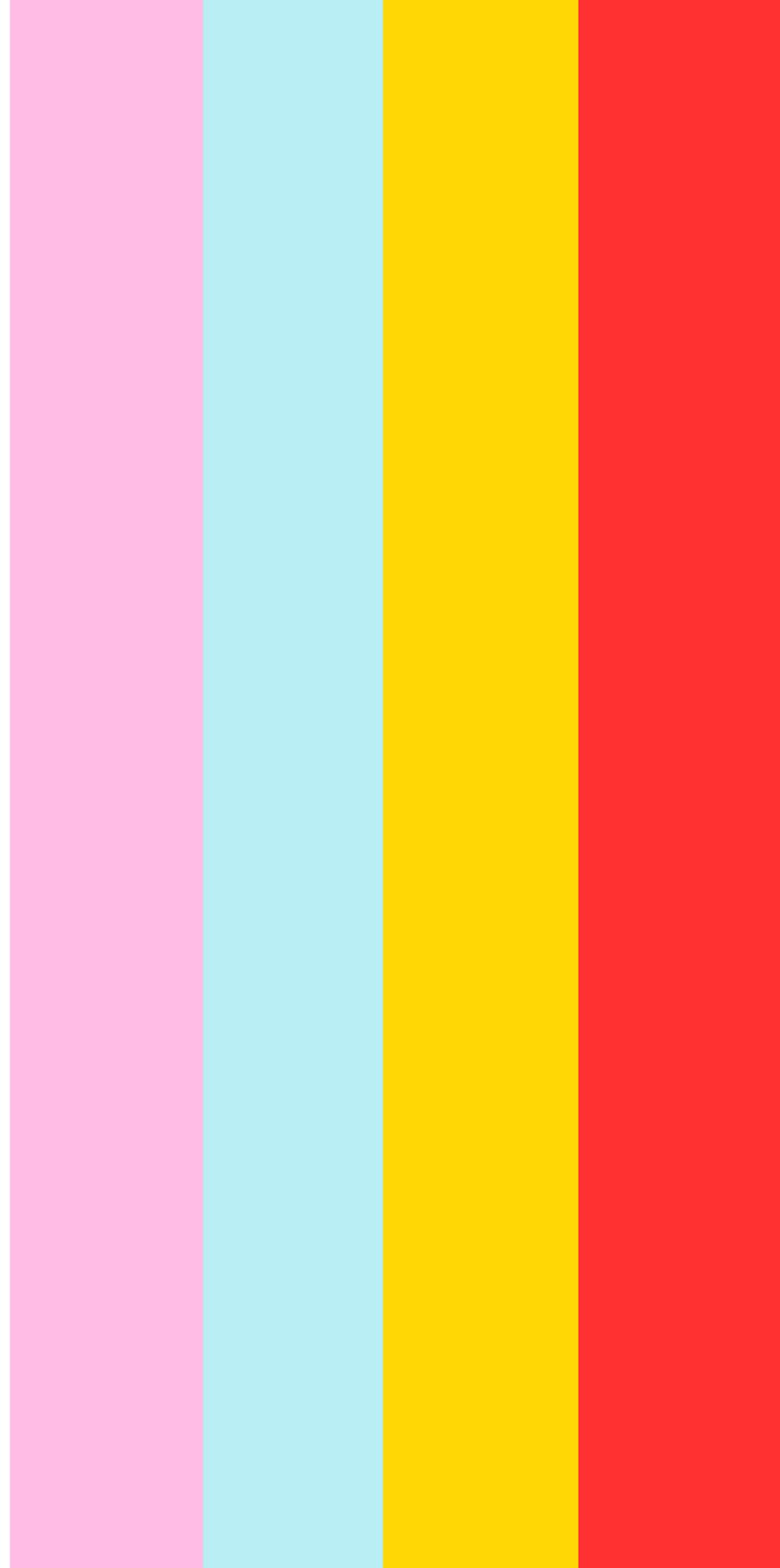


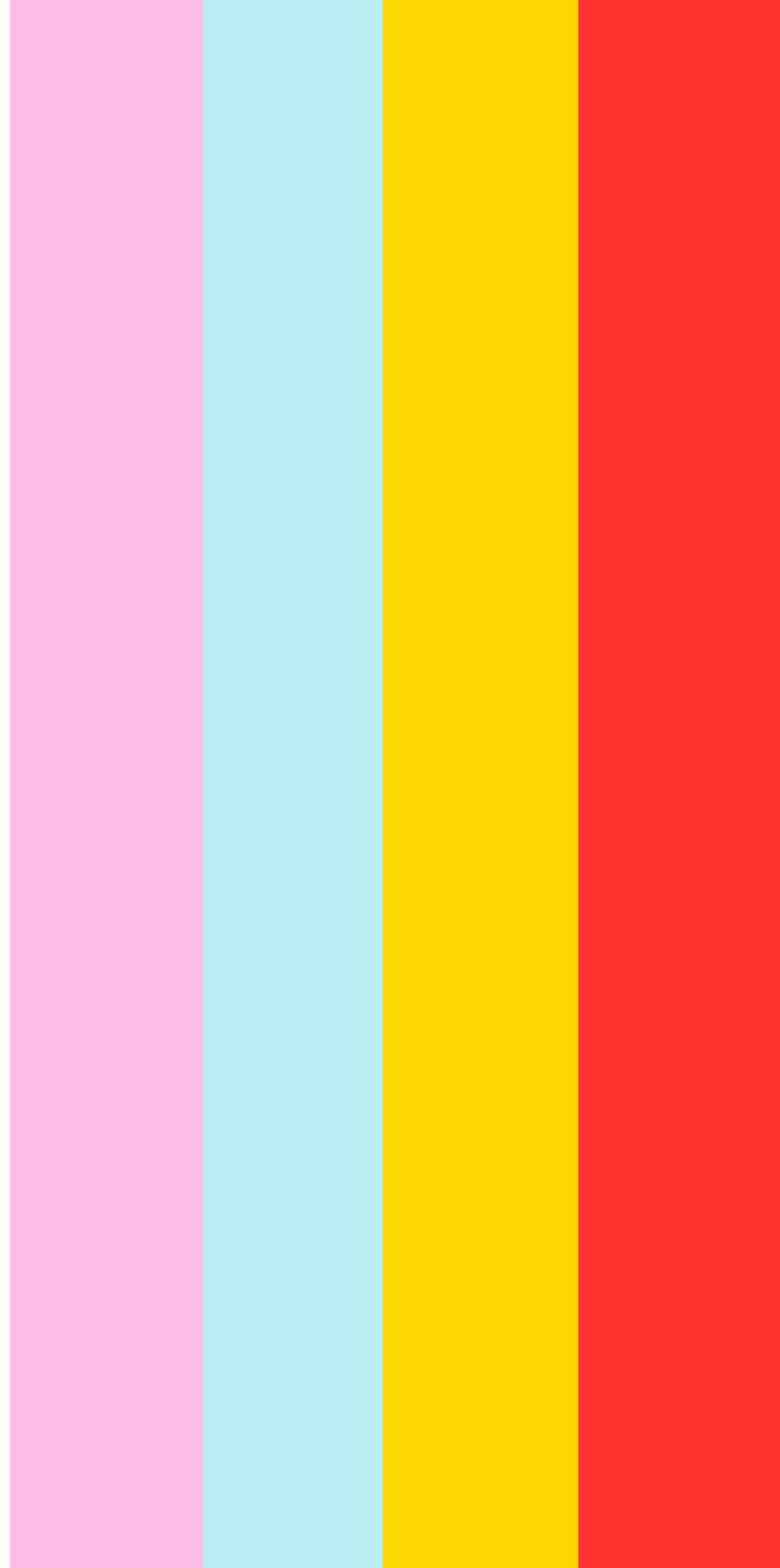
Program your GPIO pins

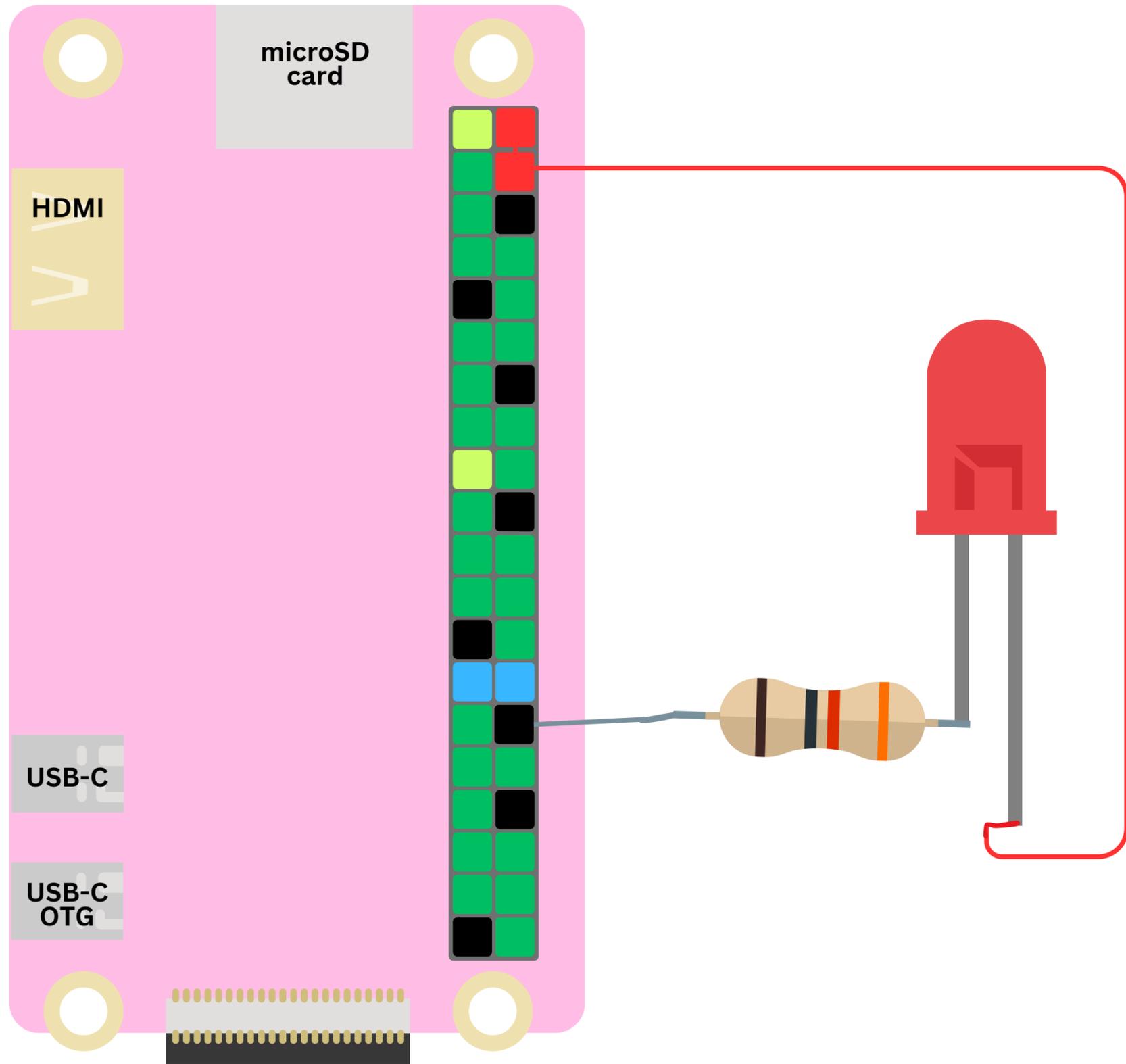
Written by Ishita Gupta



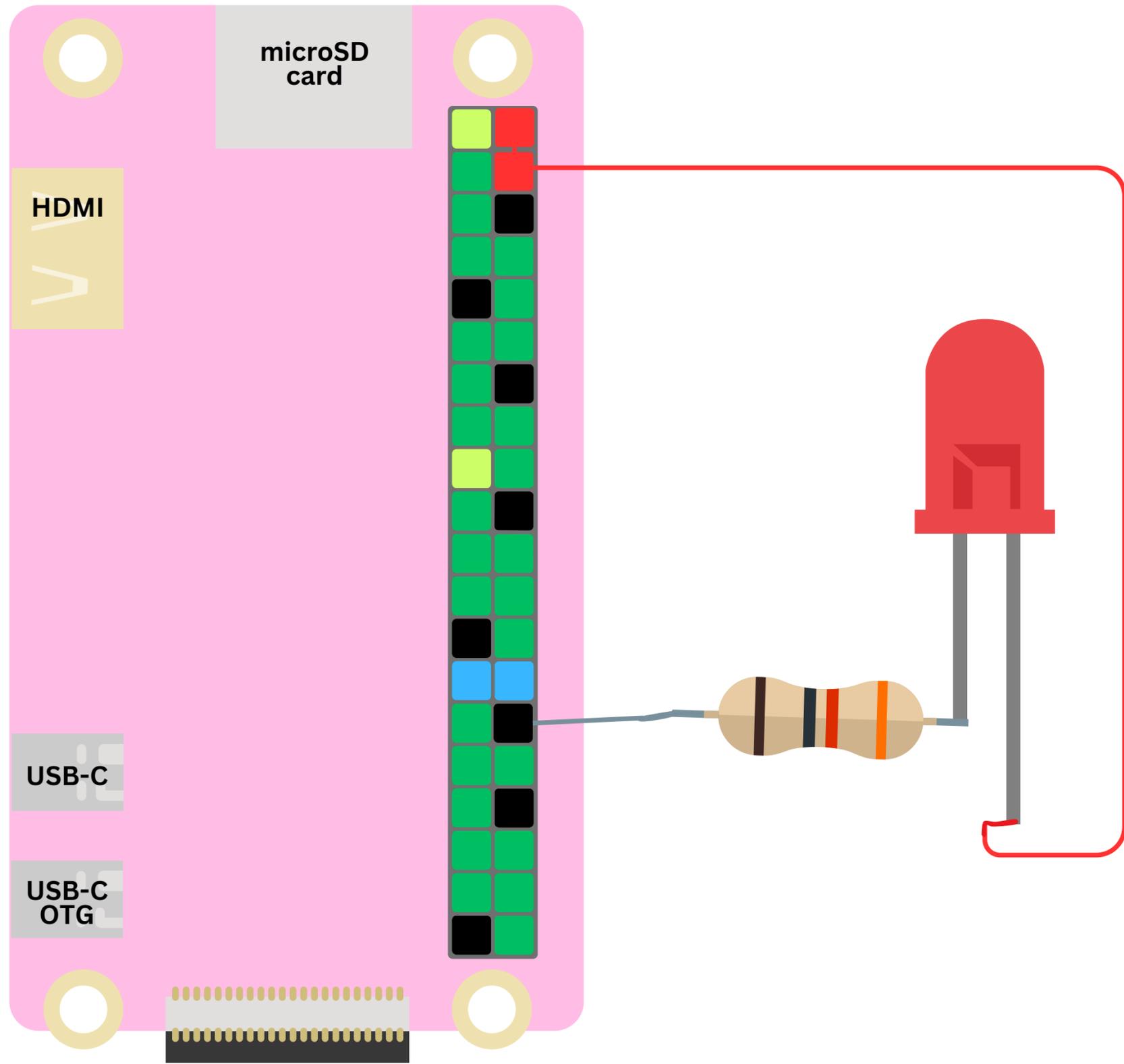
Program your GPIO pins

A guide to the user manual :)

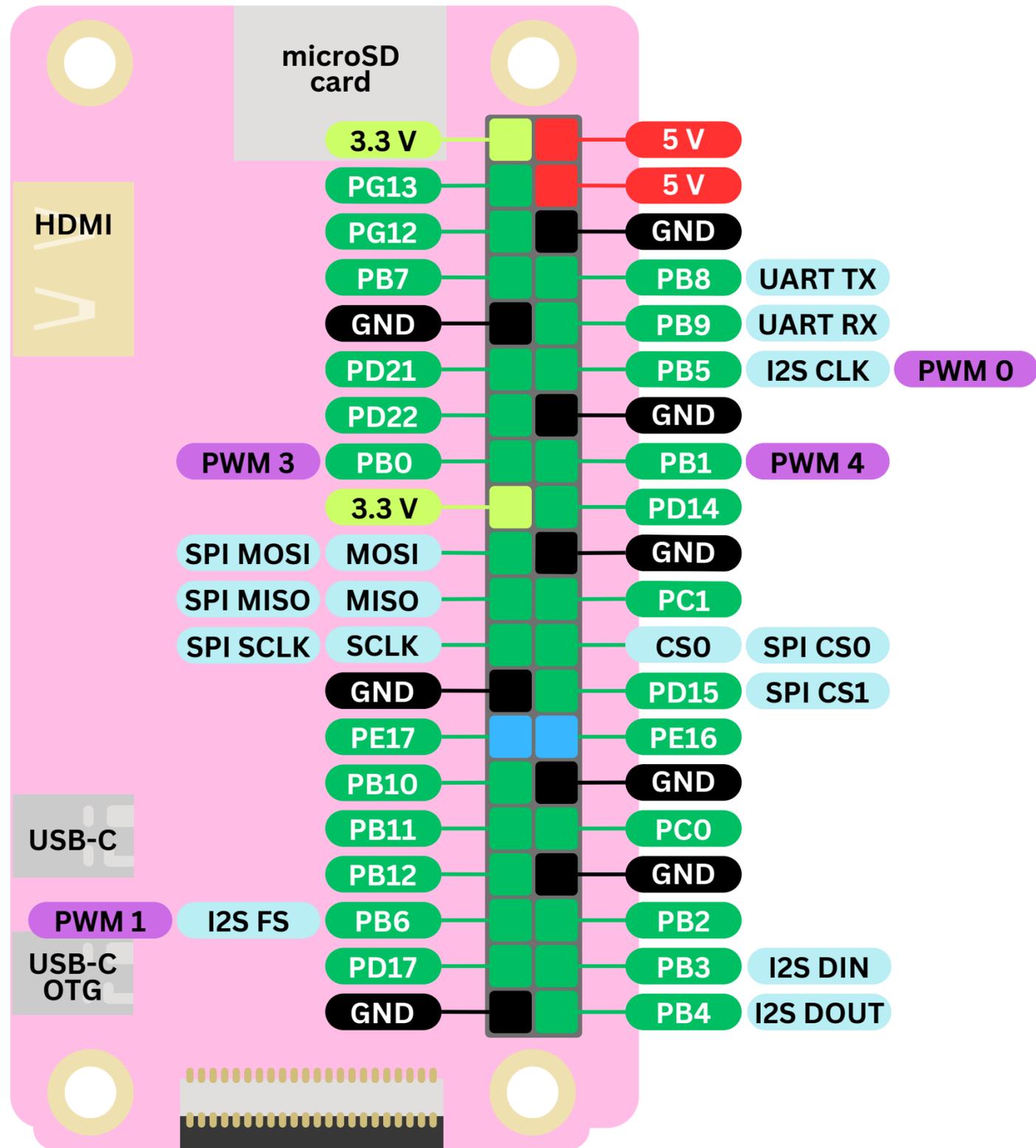




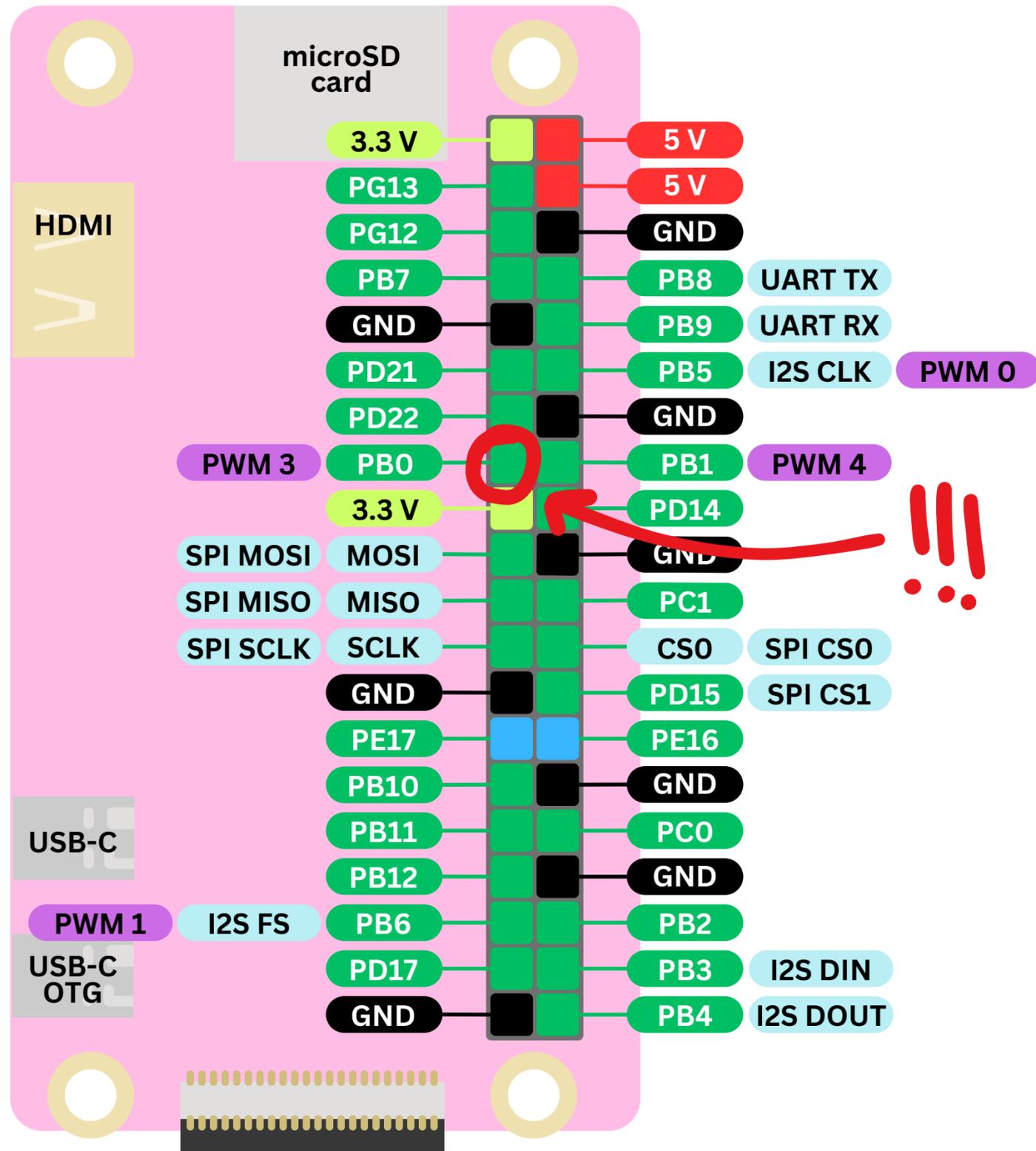
We can turn it on by connecting it to the ground and power pins



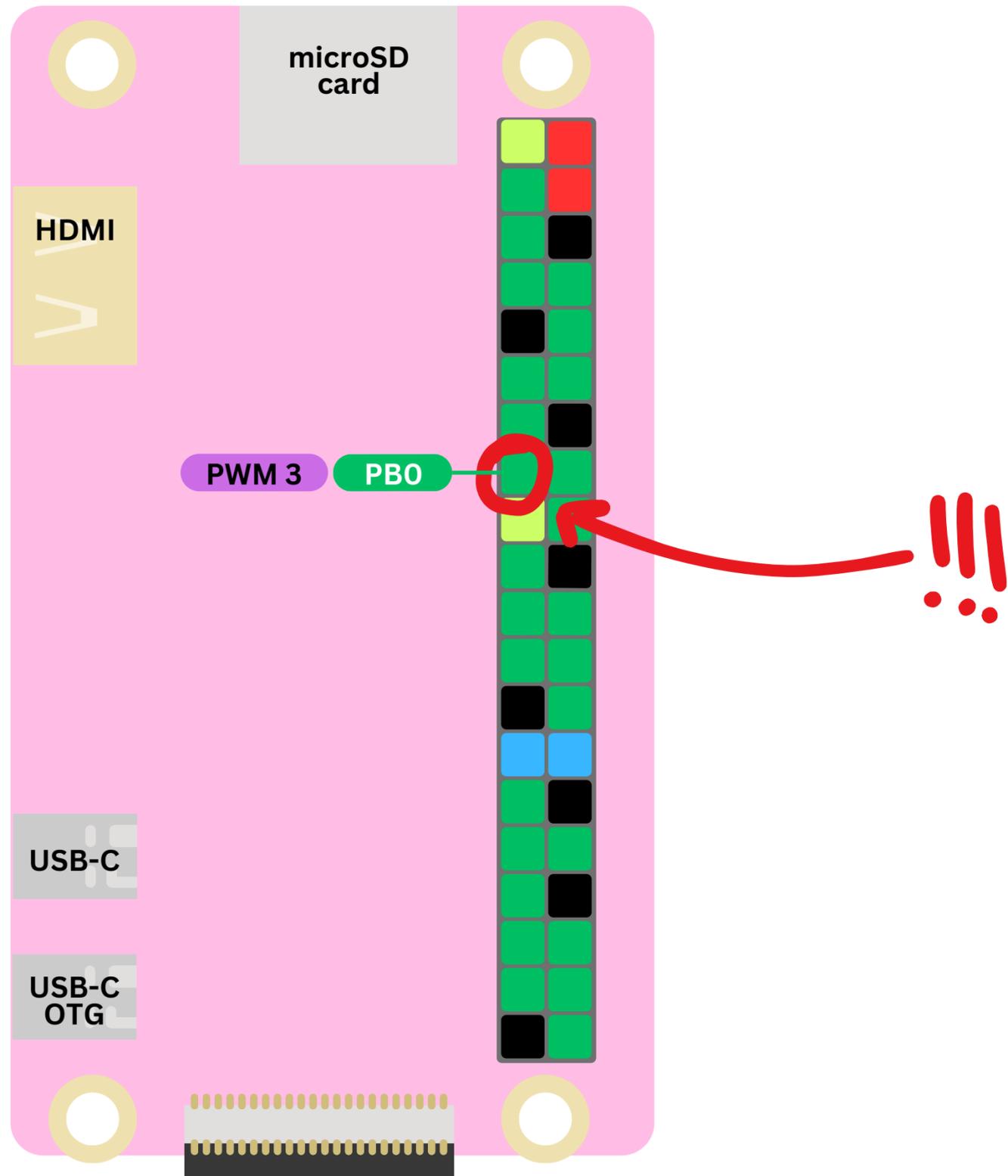
But what if we wanted to turn it on and off on demand? Using our computers?



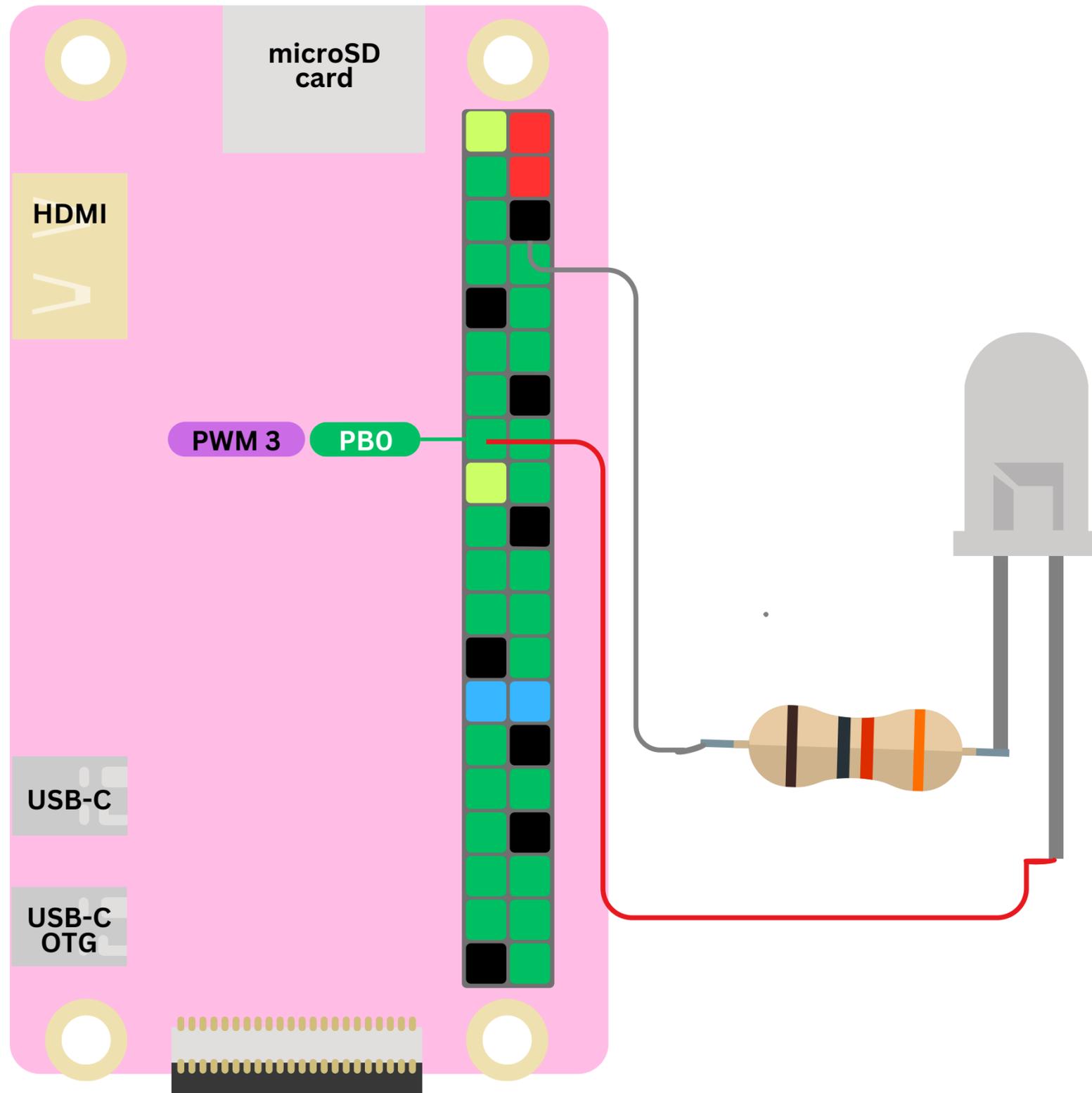
That's when all these pins come in



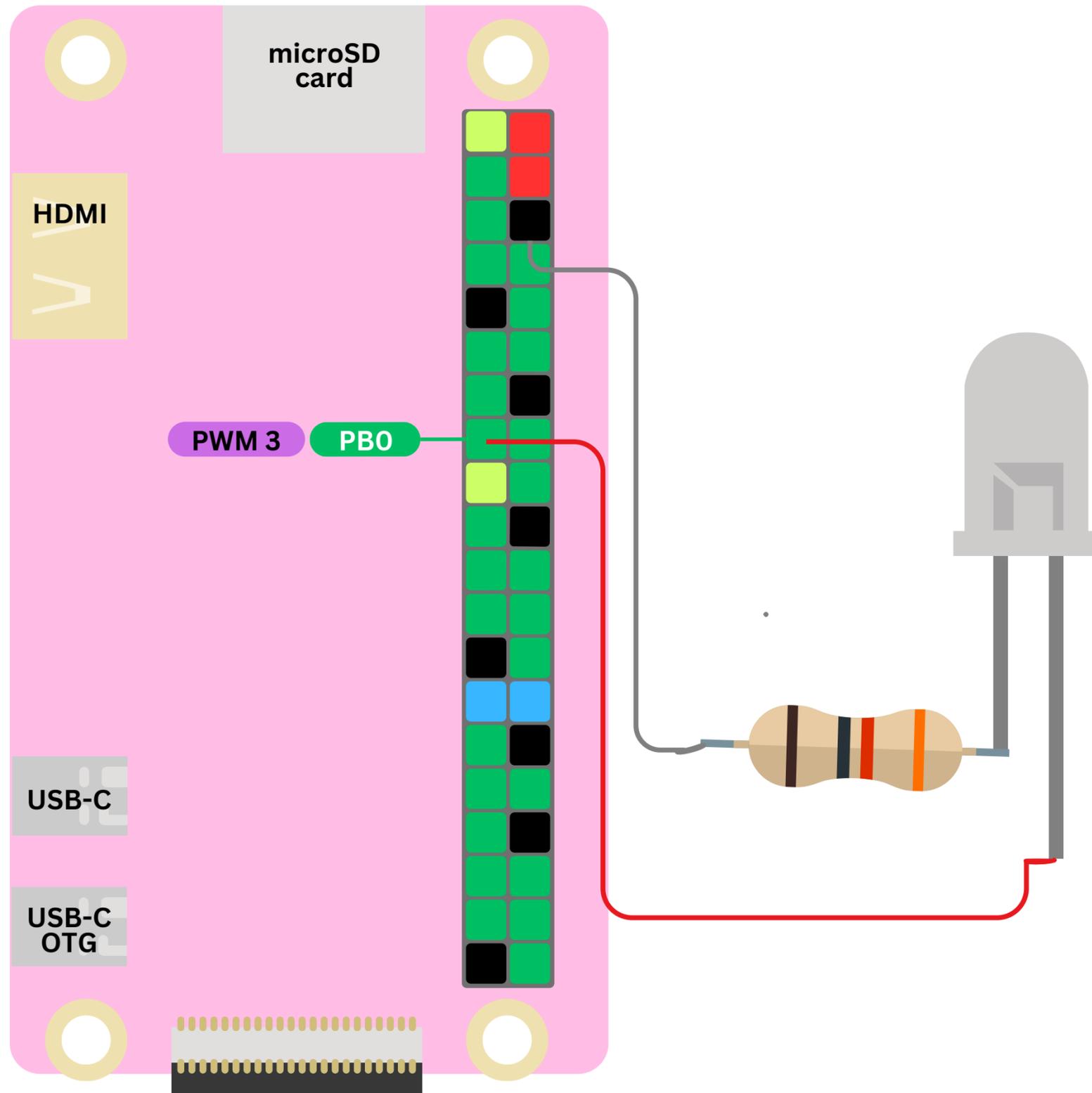
Decide which pin you want to connect to



Find its ID. The ID for this pin is PBO. This will be useful when we are telling the computer to turn this pin on.



Now connect the LED.



Now we have to program the computer to turn it on.

To program the pins, we have to do two things:

To program the pins, we have to do two things:

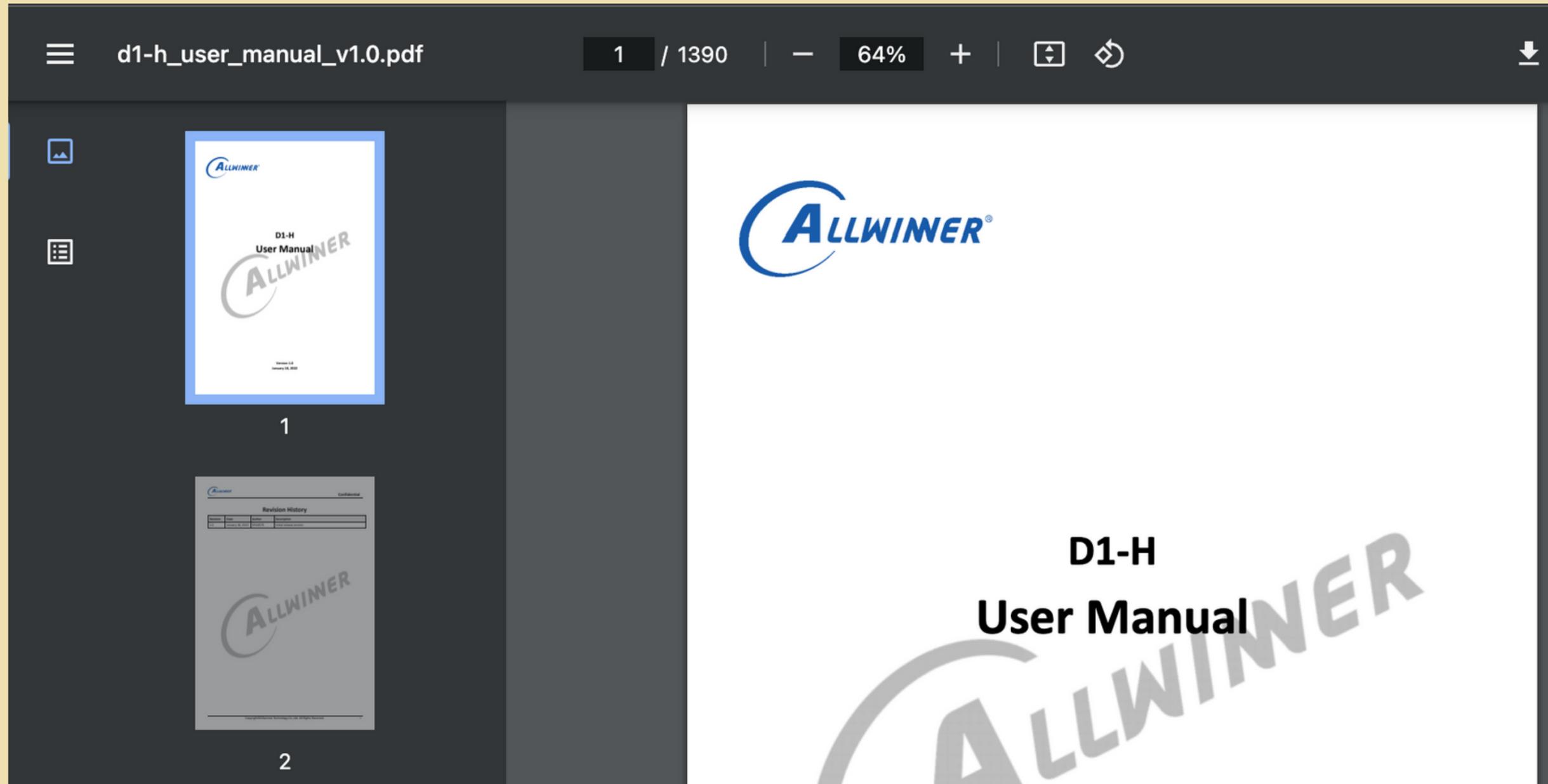
1. Tell the computer that we want the PB0 pin to be an output pin

To program the pins, we have to do two things:

1. Tell the computer that we want the PB0 pin to be an output pin
2. Tell the computer we want PB0 to be on

This is the D1-H User manual

☰ d1-h_user_manual_v1.0.pdf 1 / 1390 | - 64% + | 📄 ↺ ⬇



ALLWINNER

D1-H
User Manual

ALLWINNER

Version 1.0
January 18, 2018

1

ALLWINNER

Revision History

Version	Date	Author	Description
1.0	2018-01-18	ALLWINNER	First Release

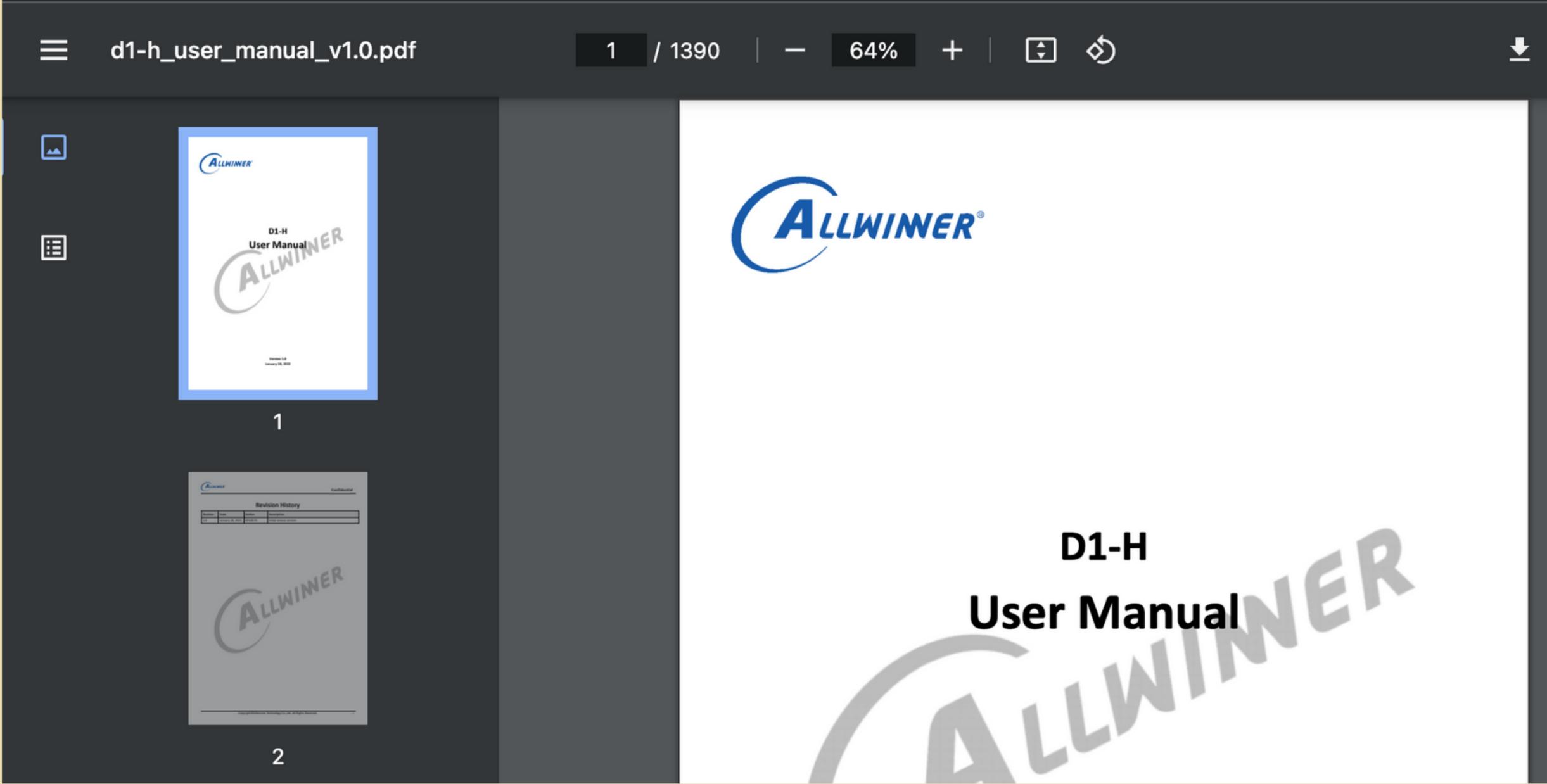
2

ALLWINNER

D1-H
User Manual

ALLWINNER

It's a MASSIVE document



It's a MASSIVE document



Navigation bar: d1-h_user_manual_v1.0.pdf | 1 / 1390 | 64% | [Zoom In] [Zoom Out] [Download]

Thumbnail 1: D1-H User Manual (Allwinner logo)

Thumbnail 2: Revision History (Allwinner logo)

Main View: Allwinner logo, D1-H User Manual, Allwinner logo

Rev. No.	Rev. Date	Rev. Description
1.0	2018.08.01	Initial Release

Today we are using only one section

PDF viewer interface for `d1-h_user_manual_v1.0.pdf`. The document is at page 1093 of 1390, zoomed to 64%. The main content area displays the **9.7.4 Register List** section, which includes the Allwinner logo, a **Confidential** watermark, and two tables.

9.7.4 Register List

Module Name	Base Address
GPIO	0x02000000

Register Name	Offset	Description
PB_CFG0	0x0030	PB Configure Register 0
PB_CFG1	0x0034	PB Configure Register 1
PB_DAT	0x0040	PB Data Register
PB_DRV0	0x0044	PB Multi_Driving Register 0
PB_DRV1	0x0048	PB Multi_Driving Register 1
PB_PULL0	0x0054	PB Pull Register 0
PC_CFG0	0x0060	PC Configure Register 0
PC_DAT	0x0070	PC Data Register
PC_DRV0	0x0074	PC Multi_Driving Register 0
PC_PULL0	0x0084	PC Pull Register 0
PD_CFG0	0x0090	PD Configure Register 0
PD_CFG1	0x0094	PD Configure Register 1
PD_CFG2	0x0098	PD Configure Register 2
PD_DAT	0x00A0	PD Data Register
PD_DRV0	0x00A4	PD Multi_Driving Register 0
PD_DRV1	0x00A8	PD Multi_Driving Register 1
PD_DRV2	0x00AC	PD Multi_Driving Register 2
PD_PULL0	0x00B4	PD Pull Register 0
PD_PULL1	0x00B8	PD Pull Register 1

Turn to this page



PDF viewer interface for 'd1-h_user_manual_v1.0.pdf'. The top bar shows page 1093 of 1390, zoomed at 64%. The left sidebar shows thumbnails for pages 1091, 1092, and 1093. The main content area displays page 1093, which contains the '9.7.4 Register List' section.

ALLWINNER Confidential

9.7.4 Register List

Module Name	Base Address
GPIO	0x02000000

Register Name	Offset	Description
PB_CFG0	0x0030	PB Configure Register 0
PB_CFG1	0x0034	PB Configure Register 1
PB_DAT	0x0040	PB Data Register
PB_DRV0	0x0044	PB Multi_Driving Register 0
PB_DRV1	0x0048	PB Multi_Driving Register 1
PB_PULL0	0x0054	PB Pull Register 0
PC_CFG0	0x0060	PC Configure Register 0
PC_DAT	0x0070	PC Data Register
PC_DRV0	0x0074	PC Multi_Driving Register 0
PC_PULL0	0x0084	PC Pull Register 0
PD_CFG0	0x0090	PD Configure Register 0
PD_CFG1	0x0094	PD Configure Register 1
PD_CFG2	0x0098	PD Configure Register 2
PD_DAT	0x00A0	PD Data Register
PD_DRV0	0x00A4	PD Multi_Driving Register 0
PD_DRV1	0x00A8	PD Multi_Driving Register 1
PD_DRV2	0x00AC	PD Multi_Driving Register 2
PD_PULL0	0x00B4	PD Pull Register 0
PD_PULL1	0x00B8	PD Pull Register 1

Or click on this image :)

PDF viewer interface for 'd1-h_user_manual_v1.0.pdf'. The top bar shows page 1093 of 1390 at 64% zoom. The main content area displays page 1093, which is the '9.7.4 Register List' section. The page includes the Allwinner logo, a 'Confidential' watermark, and a table of registers. The left sidebar shows thumbnails for pages 1091 and 1092. The right sidebar contains navigation icons for play and bookmark.

9.7.4 Register List

Module Name	Base Address
GPIO	0x02000000

Register Name	Offset	Description
PB_CFG0	0x0030	PB Configure Register 0
PB_CFG1	0x0034	PB Configure Register 1
PB_DAT	0x0040	PB Data Register
PB_DRV0	0x0044	PB Multi_Driving Register 0
PB_DRV1	0x0048	PB Multi_Driving Register 1
PB_PULL0	0x0054	PB Pull Register 0
PC_CFG0	0x0060	PC Configure Register 0
PC_DAT	0x0070	PC Data Register
PC_DRV0	0x0074	PC Multi_Driving Register 0
PC_PULL0	0x0084	PC Pull Register 0
PD_CFG0	0x0090	PD Configure Register 0
PD_CFG1	0x0094	PD Configure Register 1
PD_CFG2	0x0098	PD Configure Register 2
PD_DAT	0x00A0	PD Data Register
PD_DRV0	0x00A4	PD Multi_Driving Register 0
PD_DRV1	0x00A8	PD Multi_Driving Register 1
PD_DRV2	0x00AC	PD Multi_Driving Register 2
PD_PULL0	0x00B4	PD Pull Register 0
PD_PULL1	0x00B8	PD Pull Register 1

9.7.4 Register List

Module Name	Base Address
GPIO	0x02000000

Register Name	Offset	Description
PB_CFG0	0x0030	PB Configure Register 0
PB_CFG1	0x0034	PB Configure Register 1
PB_DAT	0x0040	PB Data Register
PB_DRV0	0x0044	PB Multi_Driving Register 0
PB_DRV1	0x0048	PB Multi_Driving Register 1
PB_PULL0	0x0054	PB Pull Register 0
PC_CFG0	0x0060	PC Configure Register 0
PC_DAT	0x0070	PC Data Register
PC_DRV0	0x0074	PC Multi_Driving Register 0
PC_PULL0	0x0084	PC Pull Register 0
PD_CFG0	0x0090	PD Configure Register 0

Confused?

WIKI

9.7.4 Register List

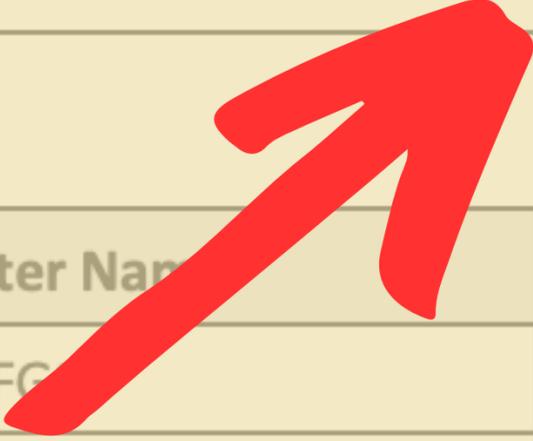
Module Name	Base Address
GPIO	0x02000000

Register Name	Offset	Description
PB_CFG0	0x0030	PB Configure Register 0
PB_CFG1	0x0034	PB Configure Register 1
PB_DAT	0x0040	PB Data Register
PB_DRV0	0x0044	PB Multi_Driving Register 0
PB_DRV1	0x0048	PB Multi_Driving Register 1
PB_PULL0	0x0054	PB Pull Register 0
PC_CFG0	0x0060	PC Configure Register 0
PC_DAT	0x0070	PC Data Register
PC_DRV0	0x0074	PC Multi_Driving Register 0
PC_PULL0	0x0084	PC Pull Register 0
PD_CFG0	0x0090	PD Configure Register 0

Let's break it down!

9.7.4 Register List

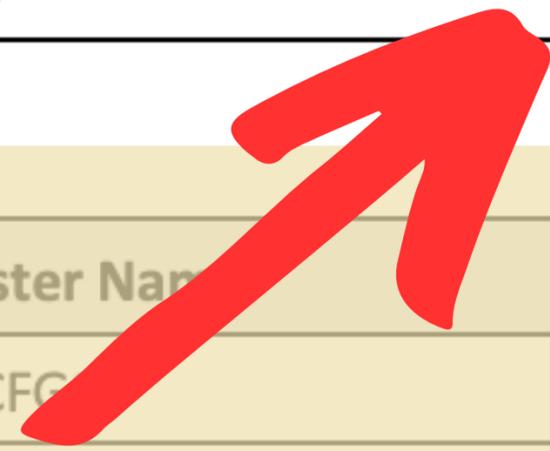
Module Name	Base Address
GPIO	0x02000000



Register Name	Offset	Description
PB_CFG	0x0030	PB Configure Register 0
PB_CFG1	0x0034	PB Configure Register 1
PB_DAT	0x0040	PB Data Register
PB_DRV0	0x0044	PB Multi_Driving Register 0
PB_DRV1	0x0048	PB Multi_Driving Register 1
PB_PULL0	0x0054	PB Pull Register 0
PC_CFG0	0x0060	PC Configure Register 0
PC_DAT	0x0070	PC Data Register
PC_DRV0	0x0074	PC Multi_Driving Register 0
PC_PULL0	0x0084	PC Pull Register 0
PD_CFG0	0x0090	PD Configure Register 0

Register List

Module Name	Base Address
GPIO	0x02000000

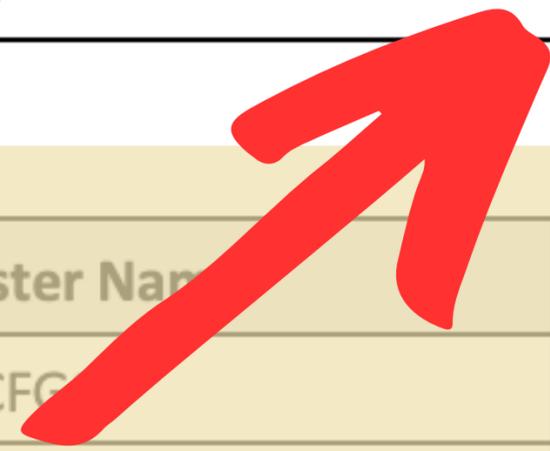


Register Name	Offset	Description
PB_CFG	0x0030	PB Configure Register 0
PB_CFG1	0x0034	PB Configure Register 1
PB_DAT	0x0040	PB Data Register
PB_DRV0	0x0044	PB Multi_Driving Register 0
PB_DRV1	0x0048	PB Multi_Driving Register 1
PB_PULL0	0x0054	PB Pull Register 0
PC_CFG0	0x0060	PC Configure Register 0
PC_DAT	0x0070	PC Data Register
PC_DRV0	0x0074	PC Multi_Driving Register 0
PC_PULL0	0x0084	PC Pull Register 0
PD_CFG0	0x0090	PD Configure Register 0

If all the GPIO pins were houses, this would be the starting number of the houses

Register List

Module Name	Base Address
GPIO	0x02000000



It looks like we are in the right place. Lets try to find the specific address we want to write to

Register Name	Offset	Description
PB_CFG	0x0030	PB Configure Register 0
PB_CFG1	0x0034	PB Configure Register 1
PB_DAT	0x0040	PB Data Register
PB_DRV0	0x0044	PB Multi_Driving Register 0
PB_DRV1	0x0048	PB Multi_Driving Register 1
PB_PULL0	0x0054	PB Pull Register 0
PC_CFG0	0x0060	PC Configure Register 0
PC_DAT	0x0070	PC Data Register
PC_DRV0	0x0074	PC Multi_Driving Register 0
PC_PULL0	0x0084	PC Pull Register 0
PD_CFG0	0x0090	PD Configure Register 0

But wait, why are we writing to addresses?

But wait, why are we writing to addresses? Recall we have to:

But wait, why are we writing to addresses? Recall we have to:

1. Tell the computer that we want the PB0 pin to be an output pin
2. Tell the computer we want PB0 to be on

To 'tell' the computer anything, we have to just put a message, a certain pattern of bits in the right location in the computer's memory. The datasheet is telling us all the different locations in the memory.

9.7.4 Register List

Module Name	Base Address
GPIO	0x02000000

This section contains information about the addresses of the PB pins in memory

Register Name	Offset	Description
PB_CFG0	0x0030	PB Configure Register 0
PB_CFG1	0x0034	PB Configure Register 1
PB_DAT	0x0040	PB Data Register
PB_DRV0	0x0044	PB Multi_Driving Register 0
PB_DRV1	0x0048	PB Multi_Driving Register 1
PB_PULL0	0x0054	PB Pull Register 0
PC_CFG0	0x0060	PC Configure Register 0
PC_DAT	0x0070	PC Data Register
PC_DRV0	0x0074	PC Multi_Driving Register 0
PC_PULL0	0x0084	PC Pull Register 0
PD_CFG0	0x0090	PD Configure Register 0

Lets start with we step 1!

1. **Tell the computer that we want the PB0 pin to be an output pin**
2. Tell the computer we want PB0 to be on

9.7.4 Register List

Module Name	Base Address
GPIO	0x02000000

1. We can use the configure register to tell the computer we want PBO to be an output

Register Name	Offset	Description
PB_CFG0	0x0030	PB Configure Register 0
PB_CFG1	0x0034	PB Configure Register 1
PB_DAT	0x0040	PB Data Register
PB_DRV0	0x0044	PB Multi_Driving Register 0
PB_DRV1	0x0048	PB Multi_Driving Register 1
PB_PULL0	0x0054	PB Pull Register 0
PC_CFG0	0x0060	PC Configure Register 0
PC_DAT	0x0070	PC Data Register
PC_DRV0	0x0074	PC Multi_Driving Register 0
PC_PULL0	0x0084	PC Pull Register 0
PD_CFG0	0x0090	PD Configure Register 0

9.7.5 Register Description

9.7.5.1 0x0030 PB Configure Register 0 (Default Value: 0xFFFF_FFFF)

The 4 bit number 0001 signifies output

Offset: 0x0030			Register Name: PB_CFG0
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0xF	PB7_SELECT
			PB7 Select
			0000:Input
			0010:LCD0-D17
			0100:TWI3-SDA
			0110:LCD0-D23
			1000:CPUBIST1
			1110:PB-EINT7



0001:Output

0011:I2S2-MCLK
0101:IR-RX
0111:UART3-RX
1001:Reserved
1111:IO Disable

9.7.5 Register Description

9.7.5.1 0x0030 PB Configure Register 0 (Default Value: 0xFFFF_FFFF)

We need to put this number in the right place for PBO in the configure register

Offset: 0x0030			Register Name: PB_CFG0
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0xF	PB7_SELECT
			PB7 Select
			0000:Input
			0001:Output
			0010:LCD0-D17
			0011:I2S2-MCLK
			0100:TWI3-SDA
			0101:IR-RX
0110:LCD0-D23			
0111:UART3-RX			
1000:CPUBIST1			
1001:Reserved			
1110:PB-EINT7			
1111:IO Disable			

9.7.4 Register List

Module Name	Base Address
GPIO	0x02000000

2. We can use the data register to tell the computer to turn that specific pin on

Register Name	Offset	Description
PB_CFG0	0x0030	PB Configure Register 0
PB_CFG1	0x0034	PB Configure Register 1
PB_DAT	0x0040	PB Data Register
PB_DRV0	0x0044	PB Multi_Driving Register 0
PB_DRV1	0x0048	PB Multi_Driving Register 1
PB_PULL0	0x0054	PB Pull Register 0
PC_CFG0	0x0060	PC Configure Register 0
PC_DAT	0x0070	PC Data Register
PC_DRV0	0x0074	PC Multi_Driving Register 0
PC_PULL0	0x0084	PC Pull Register 0
PD_CFG0	0x0090	PD Configure Register 0

9.7.4 Register List

Module Name	Base Address
GPIO	0x02000000

You will learn about the other registers later

Register Name	Offset	Description
PB_CFG0	0x0030	PB Configure Register 0
PB_CFG1	0x0034	PB Configure Register 1
PB_DAT	0x0040	PB Data Register
PB_DRV0	0x0044	PB Multi_Driving Register 0
PB_DRV1	0x0048	PB Multi_Driving Register 1
PB_PULL0	0x0054	PB Pull Register 0
PC_CFG0	0x0060	PC Configure Register 0
PC_DAT	0x0070	PC Data Register
PC_DRV0	0x0074	PC Multi_Driving Register 0
PC_PULL0	0x0084	PC Pull Register 0
PD_CFG0	0x0090	PD Configure Register 0

9.7.4 Register List

Module Name	Base Address
GPIO	0x02000000

Underneath are PC registers, which we don't need to look since we did not choose to connect to a PC pin.

Register Name	Offset	Description
PB_CFG0	0x0030	PB Configure Register 0
PB_CFG1	0x0034	PB Configure Register 1
PB_DAT	0x0040	PB Data Register
PB_DRV0	0x0044	PB Multi_Driving Register 0
PB_DRV1	0x0048	PB Multi_Driving Register 1
PB_PULL0	0x0054	PB Pull Register 0
PC_CFG0	0x0060	PC Configure Register 0
PC_DAT	0x0070	PC Data Register
PC_DRV0	0x0074	PC Multi_Driving Register 0
PC_PULL0	0x0084	PC Pull Register 0
PD_CFG0	0x0090	PD Configure Register 0

9.7.4 Register List

Module Name	Base Address
GPIO	0x02000000

Similarly if we had connected to a PD pin, we could look down here, and so on.

Register Name	Offset	Description
PB_CFG0	0x0030	PB Configure Register 0
PB_CFG1	0x0034	PB Configure Register 1
PB_DAT	0x0040	PB Data Register
PB_DRV0	0x0044	PB Multi_Driving Register 0
PB_DRV1	0x0048	PB Multi_Driving Register 1
PB_PULL0	0x0054	PB Pull Register 0
PC_CFG0	0x0060	PC Configure Register 0
PC_DAT	0x0070	PC Data Register
PC_DRV0	0x0074	PC Multi_Driving Register 0
PC_PULL0	0x0084	PC Pull Register 0
PD_CFG0	0x0090	PD Configure Register 0

Register List

Module Name	Base Address
GPIO	0x02000000

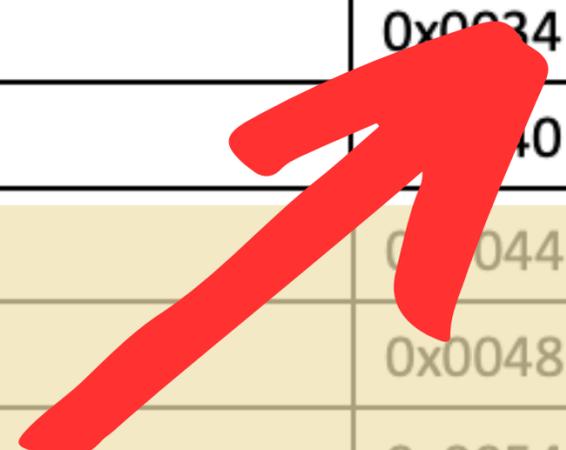
Register Name	Offset	Description
PB_CFG0	0x0030	PB Configure Register 0
PB_CFG1	0x0034	PB Configure Register 1
PB_DAT	0x0040	PB Data Register
PB_DRV0	0x0044	PB Multi_Driving Register 0
PB_DRV1	0x0048	PB Multi_Driving Register 1
PB_PULL0	0x0054	PB Pull Register 0
PC_CFG0	0x0060	PC Configure Register 0
PC_DAT	0x0070	PC Data Register
PC_DRV0	0x0074	PC Multi_Driving Register 0
PC_PULL0	0x0084	PC Pull Register 0
PD_CFG0	0x0090	PD Configure Register 0

9.7.4 Register List

Module Name	Base Address
GPIO	0x02000000

Register Name	Offset	Description
PB_CFG0	0x0030	PB Configure Register 0
PB_CFG1	0x0034	PB Configure Register 1
PB_DAT	0x0040	PB Data Register
PB_DRV0	0x0044	PB Multi_Driving Register 0
PB_DRV1	0x0048	PB Multi_Driving Register 1
PB_PULL0	0x0054	PB Pull Register 0
PC_CFG0	0x0060	PC Configure Register 0
PC_DAT	0x0070	PC Data Register
PC_DRV0	0x0074	PC Multi_Driving Register 0
PC_PULL0	0x0084	PC Pull Register 0
PD_CFG0	0x0090	PD Configure Register 0

These are all offsets



Register List

Module Name	Base Address
GPIO	0x02000000

Register Name	Offset	Description
PB_CFG0	0x0030	PB Configure Register 0
PB_CFG1	0x0034	PB Configure Register 1
PB_DAT	0x0040	PB Data Register

PB_DRV0	0x0044	PB Multi_Driving Register 0
PB_DRV1	0x0048	PB Multi_Driving Register 1
PB_PULL0	0x0054	PB Pull Register 0
PC_CFG0	0x0060	PC Configure Register 0
PC_CFG1	0x0064	PC Configure Register 1
PC_DAT	0x0070	PC Data Register
PC_DRV0	0x0074	PC Multi_Driving Register 0
PC_PULL0	0x0084	PC Pull Register 0
PD_CFG0	0x0090	PD Configure Register 0

You add the offset to the base address to access the address that you need

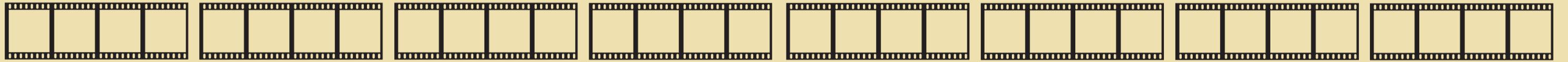
9.7.4 Register List

Module Name	Base Address
GPIO	0x02000000

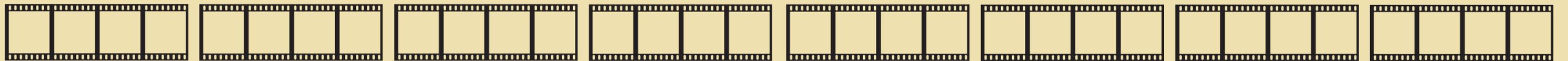
Register Name	Offset	Description
PB_CFG0	0x0030	PB Configure Register 0
PB_CFG1	0x0034	PB Configure Register 1
PB_DAT	0x0040	PB Data Register
PB_DRV0	0x0044	PB Multi_Driving Register 0
PB_DRV1	0x0048	PB Multi_Driving Register 1
PB_PULL0	0x0054	PB Pull Register 0
PC_CFG0	0x0060	PC Configure Register 0
PC_DAT	0x0070	PC Data Register
PC_DRV0	0x0074	PC Multi_Driving Register 0
PC_PULL0	0x0084	PC Pull Register 0
PD_CFG0	0x0090	PD Configure Register 0

Lets look at 0x20000030 and 0x20000034,
the two configure registers for the PB group

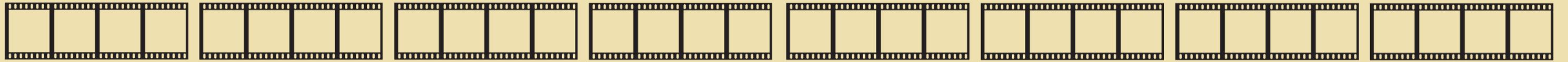
0x2000030



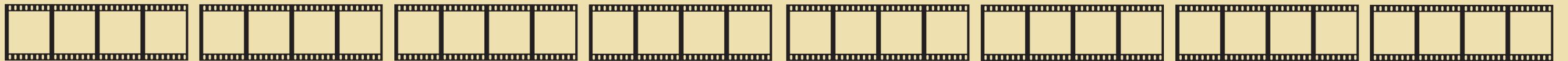
0x2000034



0x2000030

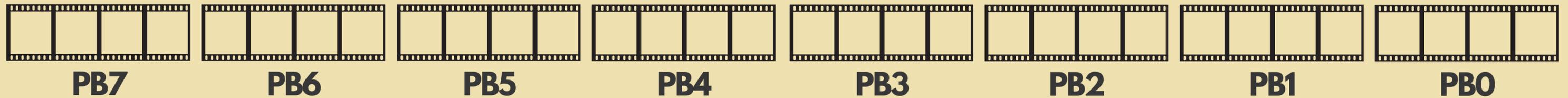


0x2000034

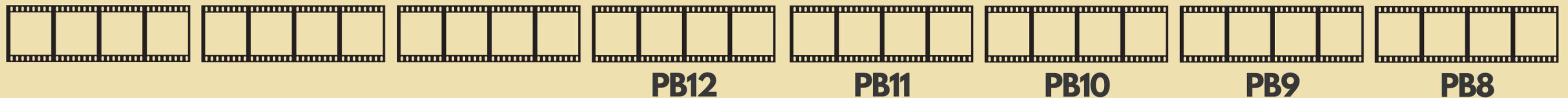


Here it is! Each box can store a 0 or a 1 bit

0x2000030

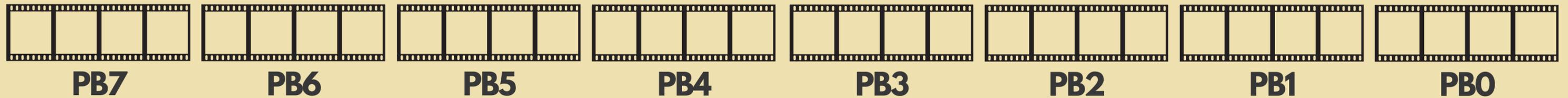


0x2000034

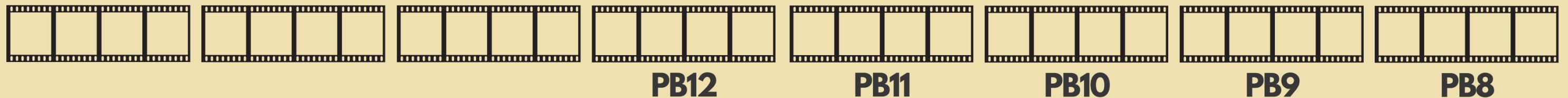


Here is how the space is allocated. 4 bits for each pin.

0x2000030

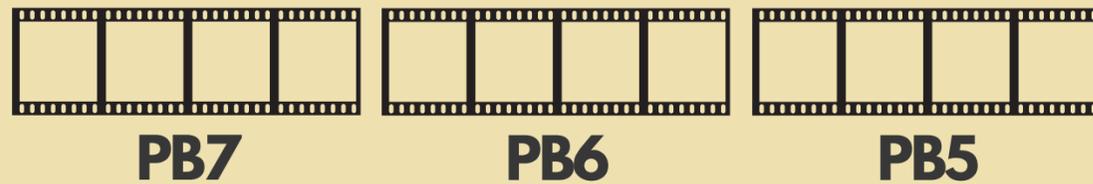


0x2000034



Here is how the space is allocated. 4 bits for each pin. This makes sense, Recall:

0x2000030



0x2000034



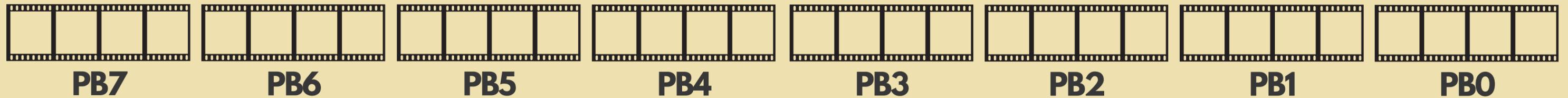
9.7.5 Register Description

9.7.5.1 0x0030 PB Configure Register 0 (Default Value: 0xFFFF_FFFF)

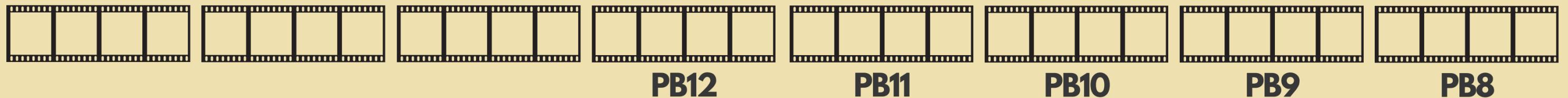
Offset: 0x0030			Register Name: PB_CFG0
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0xF	PB7_SELECT
			PB7 Select
			0000:Input
			0001:Output
			0010:LCD0-D17
			0011:I2S2-MCLK
			0100:TWI3-SDA
			0101:IR-RX
			0110:LCD0-D23
			0111:UART3-RX
			1000:CPUBIST1
			1001:Reserved
			1110:PB-EINT7
			1111:IO Disable

Here is how the space is allocated. 4 bits for each pin. This makes sense, Recall:

0x2000030

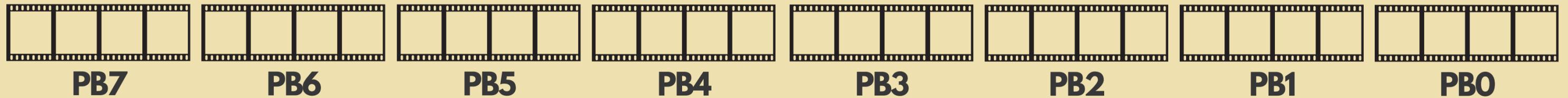


0x2000034

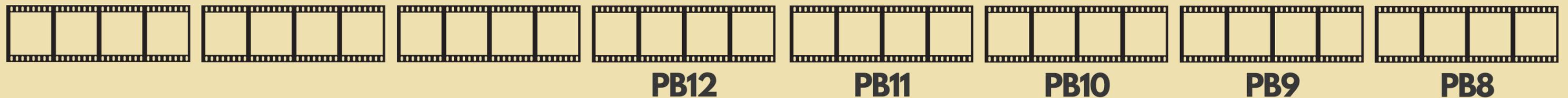


Since there are only 12 PB pins, some space is unused

0x2000030

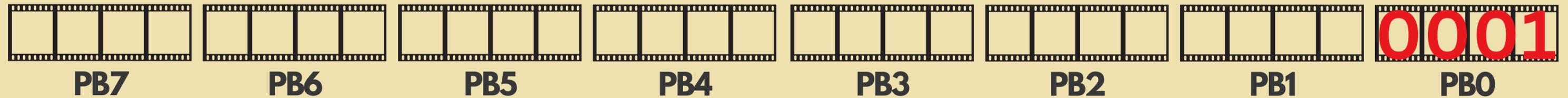


0x2000034

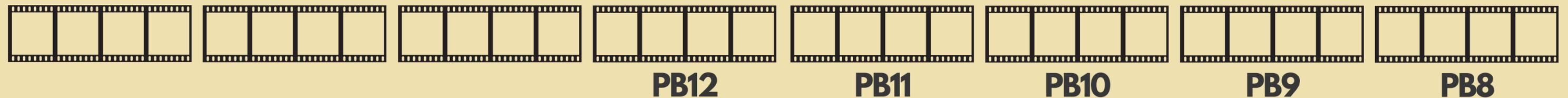


Because we want PB0 to be an output, we can put that 0001 in the right place.

0x2000030

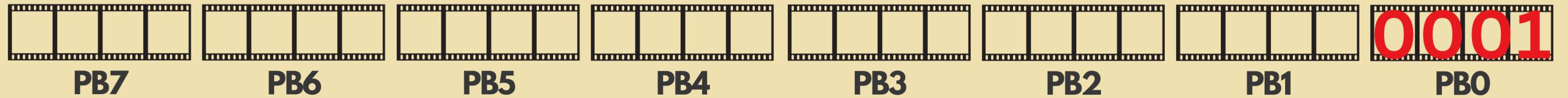


0x2000034

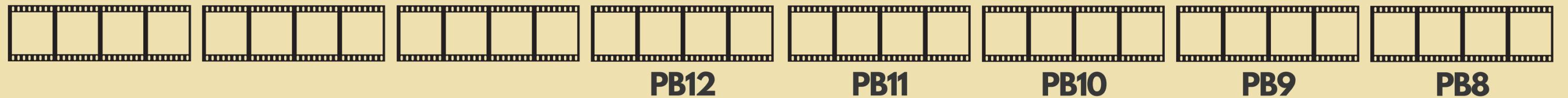


Because we want PB0 to be an output, we can put that 0001 in the right place.

0x2000030

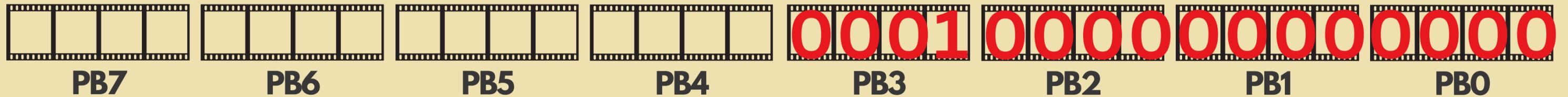


0x2000034

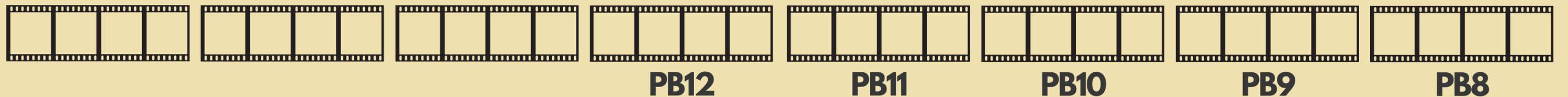


Here it is!

0x2000030



0x2000034



If we had picked PB3, we would have done this

Now we know how to do step 1!

- ~~1. Tell the computer that we want the PB0 pin to be an output pin~~
2. Tell the computer we want PB0 to be on

Lets do step 2:

1. Tell the computer that we want the PBO pin to be an output pin
2. **Tell the computer we want PBO to be on**

9.7.4 Register List

Module Name	Base Address
GPIO	0x02000000

2. We can use the data register to tell the computer to turn that specific pin on

Register Name	Offset	Description
PB_CFG0	0x0030	PB Configure Register 0
PB_CFG1	0x0034	PB Configure Register 1
PB_DAT	0x0040	PB Data Register
PB_DRV0	0x0044	PB Multi_Driving Register 0
PB_DRV1	0x0048	PB Multi_Driving Register 1
PB_PULL0	0x0054	PB Pull Register 0
PC_CFG0	0x0060	PC Configure Register 0
PC_DAT	0x0070	PC Data Register
PC_DRV0	0x0074	PC Multi_Driving Register 0
PC_PULL0	0x0084	PC Pull Register 0
PD_CFG0	0x0090	PD Configure Register 0

9.7.4 Register List

Module Name	Base Address
GPIO	0x02000000

Register Name	Offset	Description
PB_CFG0	0x0030	PB Configure Register 0
PB_CFG1	0x0034	PB Configure Register 1
PB_DAT	0x0040	PB Data Register
PB_DRV0	0x0044	PB Multi_Driving Register 0
PB_DRV1	0x0048	PB Multi_Driving Register 1
PB_PULL0	0x0054	PB Pull Register 0
PC_CFG0	0x0060	PC Configure Register 0
PC_DAT	0x0070	PC Data Register
PC_DRV0	0x0074	PC Multi_Driving Register 0
PC_PULL0	0x0084	PC Pull Register 0
PD_CFG0	0x0090	PD Configure Register 0

Add the offset to the base address, to go to the data register

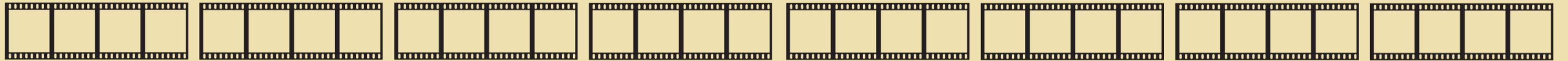
9.7.4 Register List

Module Name	Base Address
GPIO	0x02000000

Register Name	Offset	Description
PB_CFG0	0x0030	PB Configure Register 0
PB_CFG1	0x0034	PB Configure Register 1
PB_DAT	0x0040	PB Data Register
PB_DRV0	0x0044	PB Multi_Driving Register 0
PB_DRV1	0x0048	PB Multi_Driving Register 1
PB_PULL0	0x0054	PB Pull Register 0
PC_CFG0	0x0060	PC Configure Register 0
PC_CFG1	0x0064	PC Configure Register 1
PC_DAT	0x0070	PC Data Register
PC_DRV0	0x0074	PC Multi_Driving Register 0
PC_PULL0	0x0084	PC Pull Register 0
PD_CFG0	0x0090	PD Configure Register 0

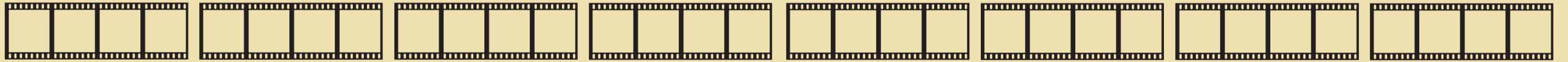
Lets go to 0x02000040

0x2000040



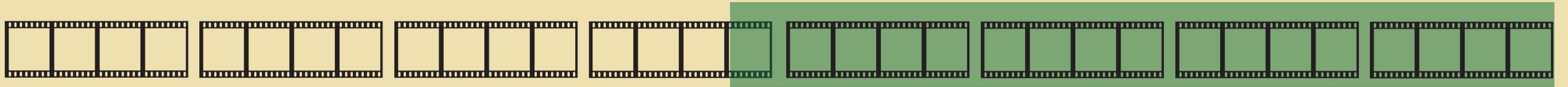
Space is allocated a bit differently here

0x2000040



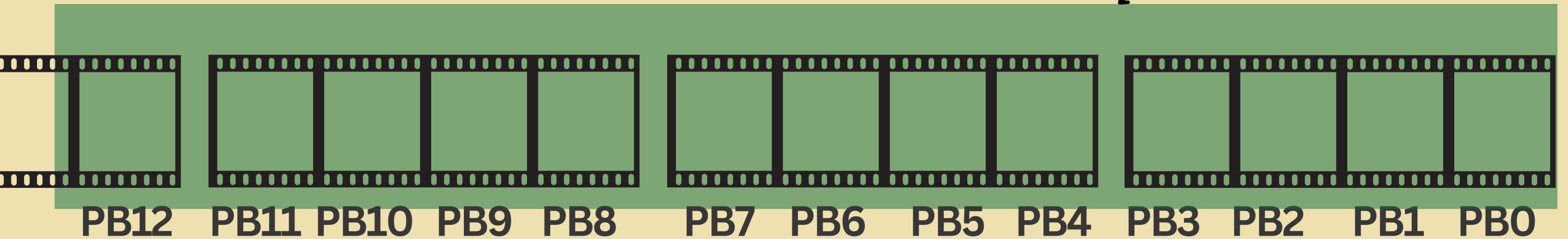
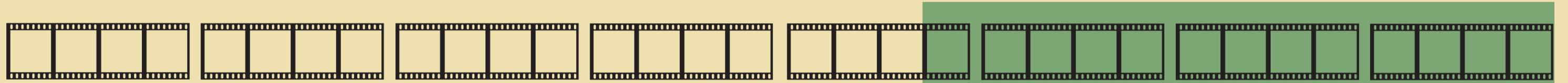
**We only need one bit to say whether the pin
needs to be on or off**

0x2000040



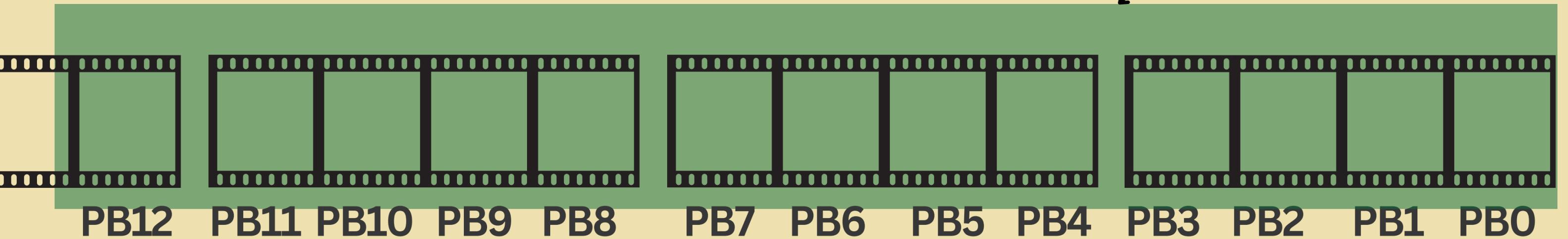
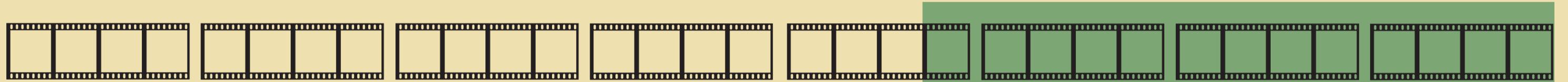
So we only use these 13 bits for the 12 PB pins

0x2000040



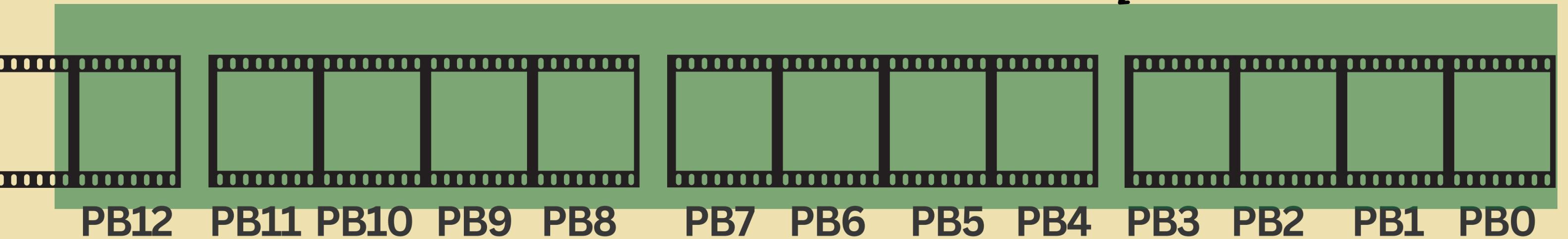
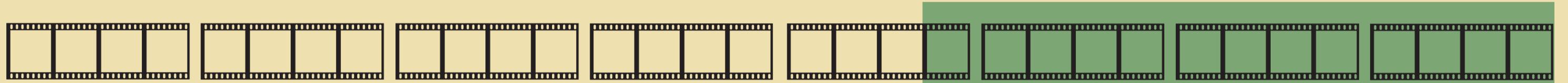
Lets take a closer look

0x2000040



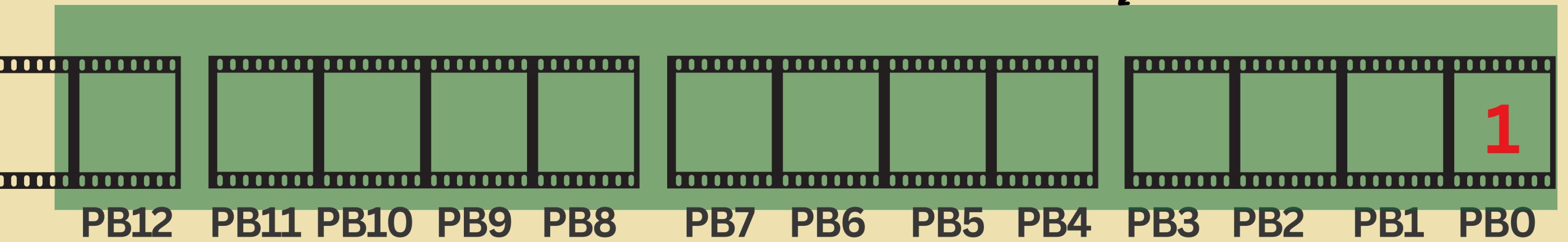
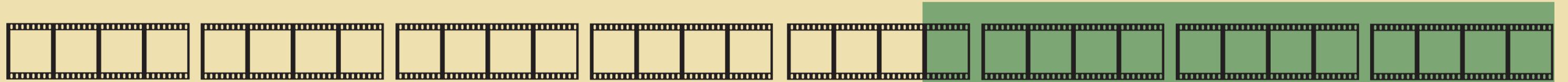
One bit for each pin!

0x200040

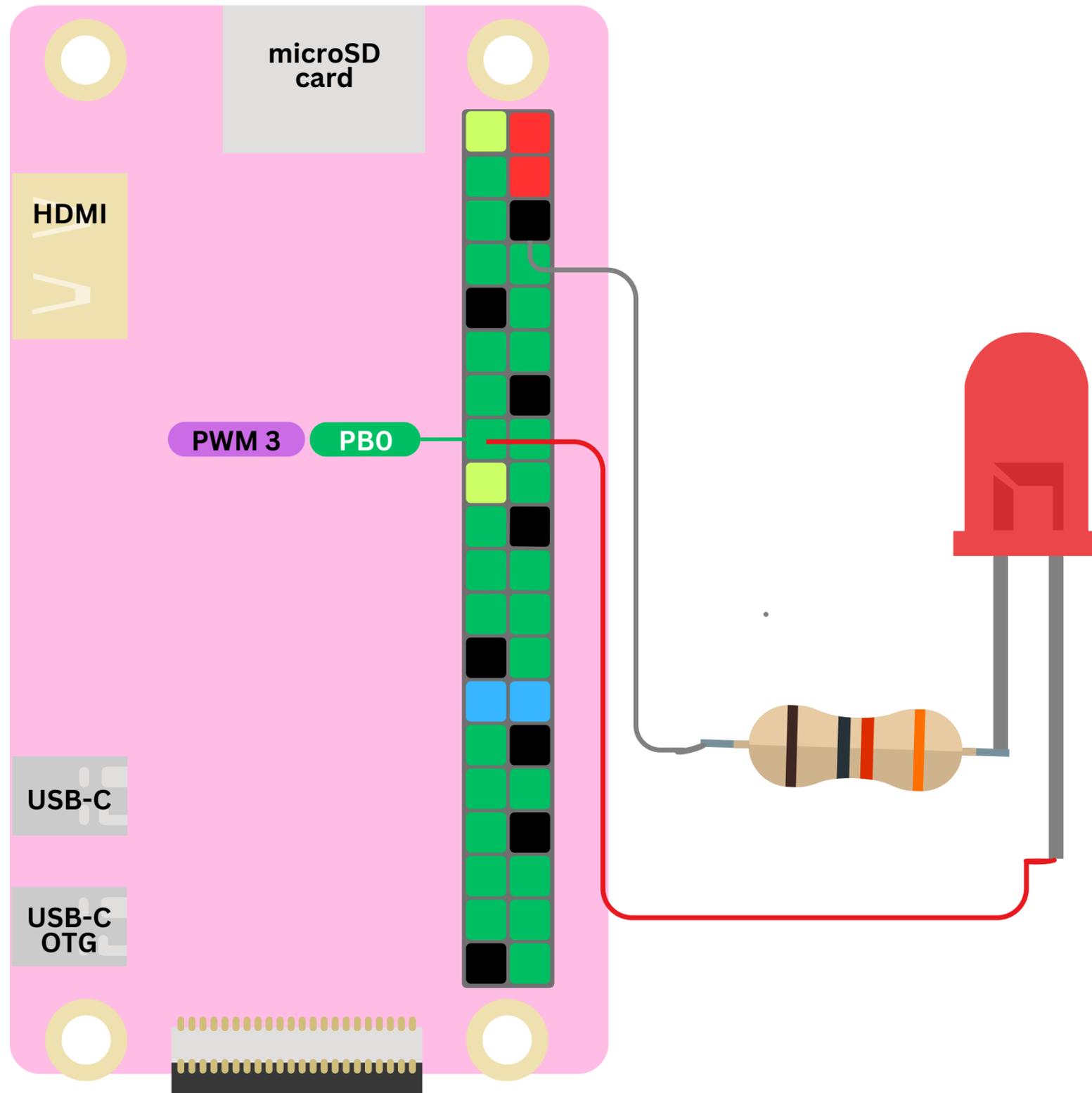


Just add 1 to the location at PB0!

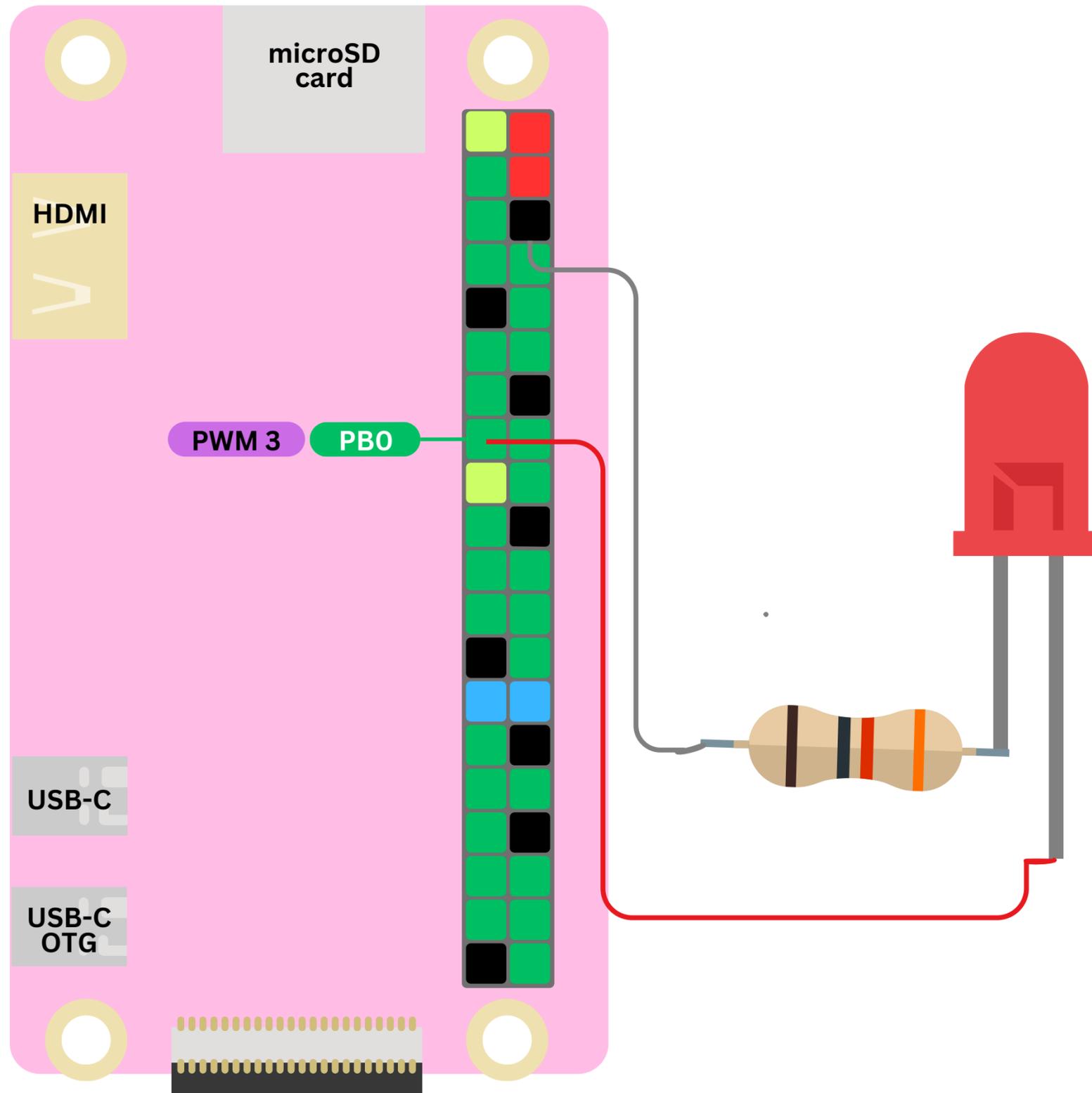
0x2000040



Just add 1 to the location at PB0!



And we are done!



Now you understand how to use the datasheet to program GPIO pins!