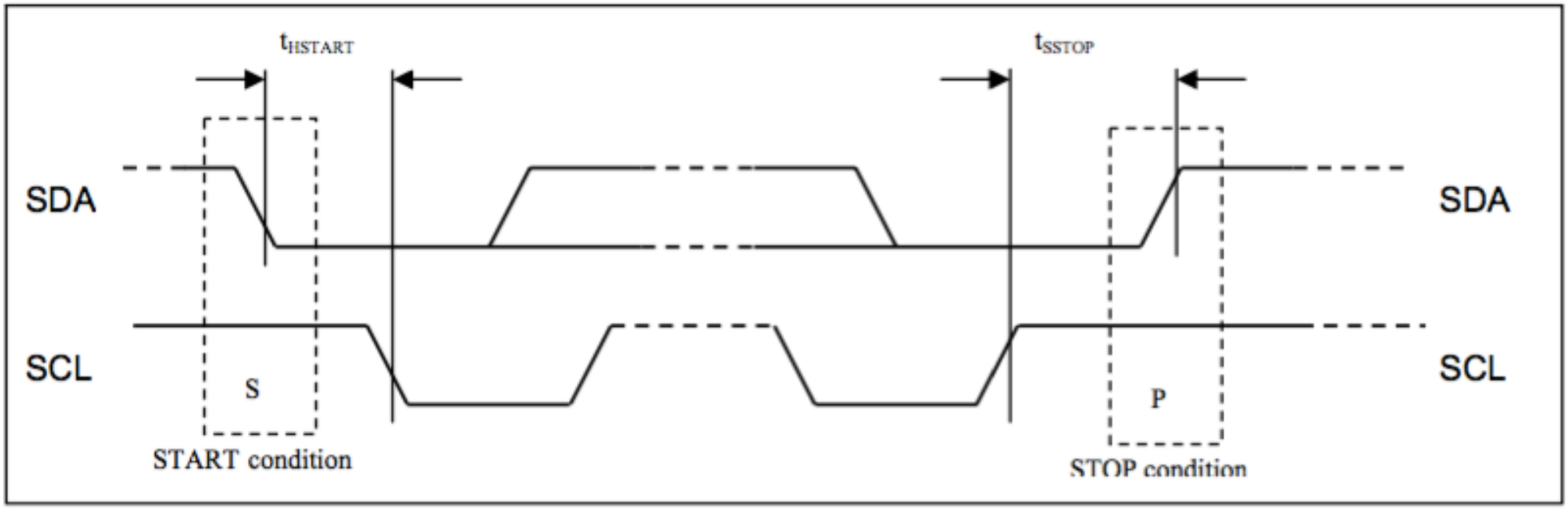
I²C BUS

- Devices implement vendor-specific protocols for data transfer and command
  - Data sheet may report even I2C low-level specs

Figure 8-8 : Definition of the Start and Stop Condition

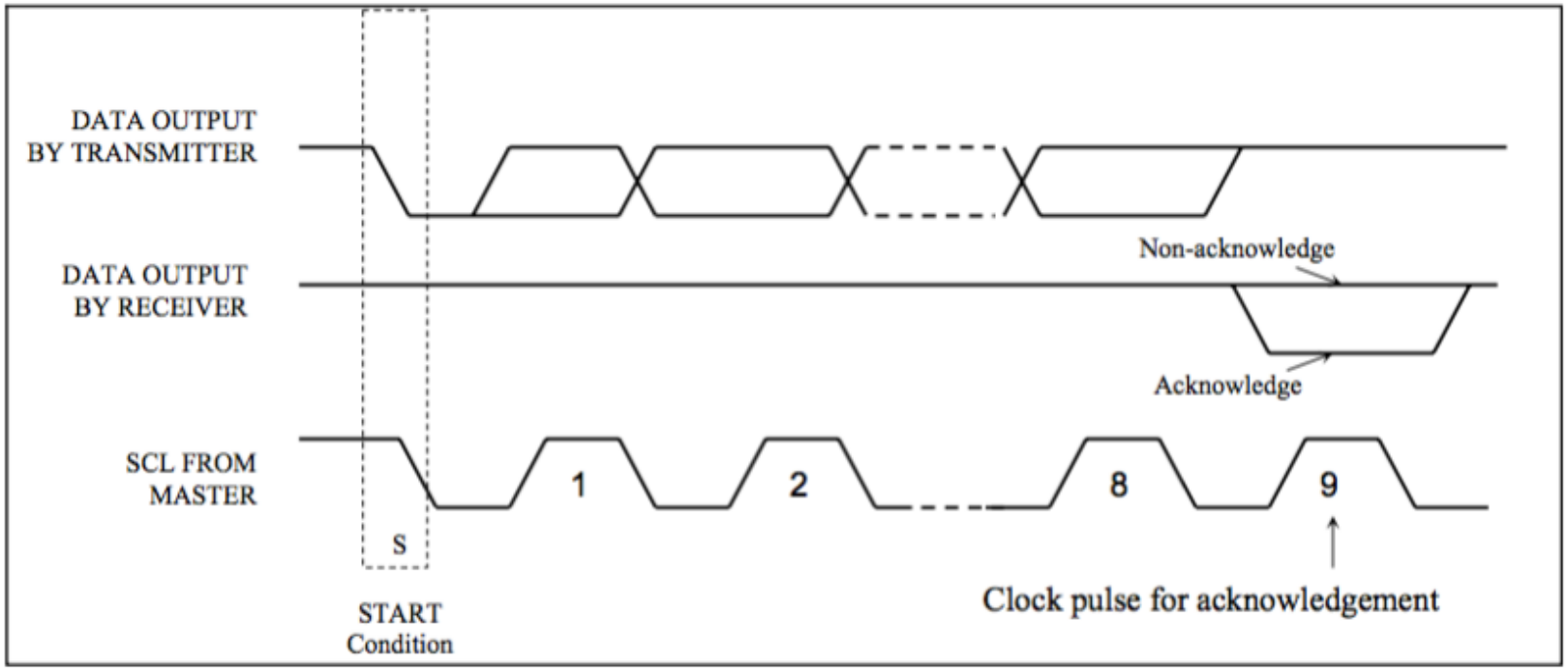


SSD1306 OLED Display Datasheet - (SSD1306 Datasheet for 096 OLED.pdf) page 21

I²C BUS

- Devices implement vendor-specific protocols for data transfer and command
  - Data sheet may report even I2C low-level specs

Figure 8-9 : Definition of the acknowledgement condition

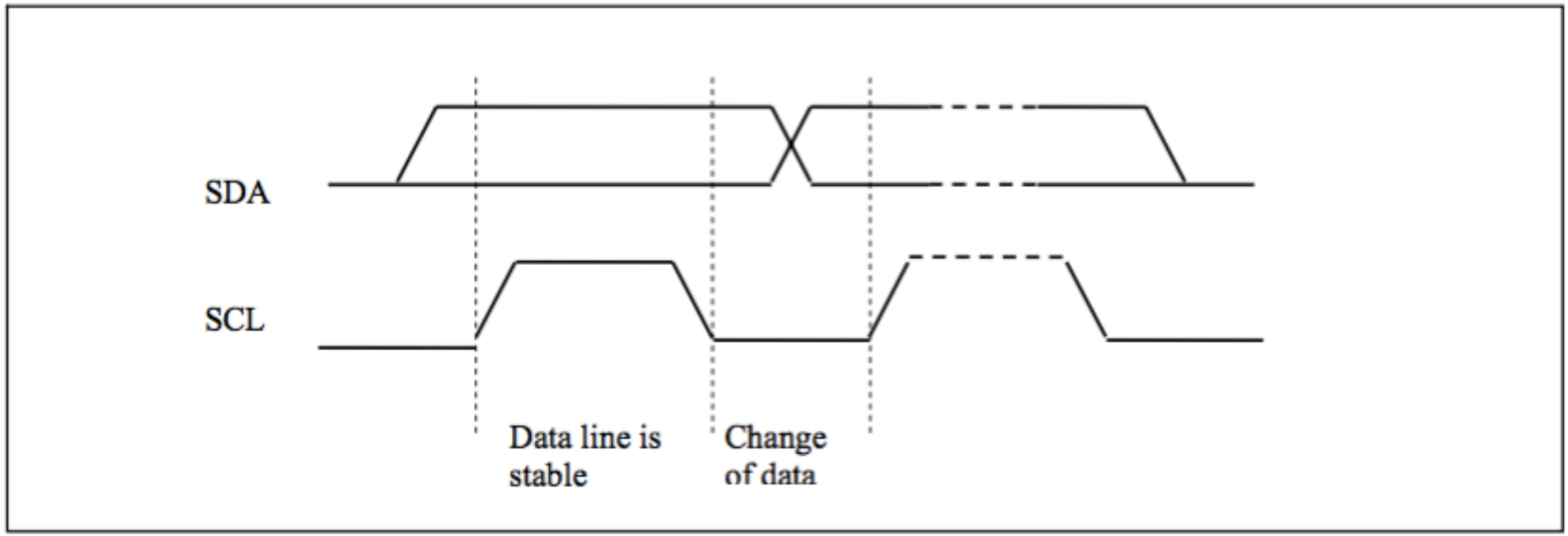


SSD1306 OLED Display Datasheet - (SSD1306 Datasheet for 096 OLED.pdf) page 21

I²C BUS

- Devices implement vendor-specific protocols for data transfer and command
  - Data sheet may report even I2C low-level specs

Figure 8-10 : Definition of the data transfer condition



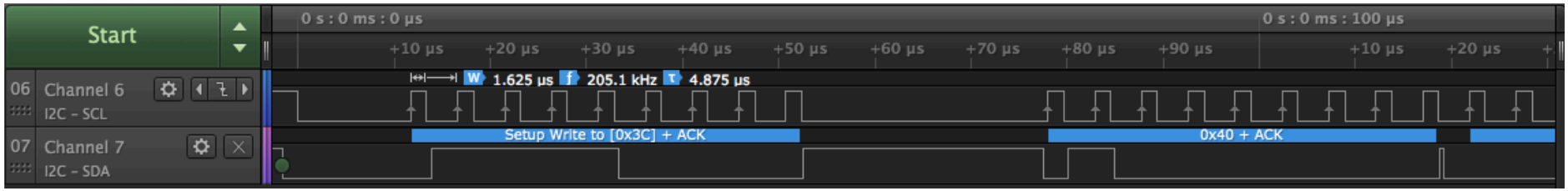
SSD1306 OLED Display Datasheet - (SSD1306 Datasheet for 096 OLED.pdf) page 21

SSD1306 OLED DISPLAY

Displaying an 8x8 pixel character

- Command Data Write ($40) is sent followed by 8 bytes, one for each pixel row

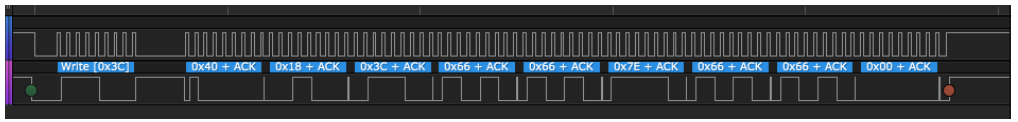
```
CHAR A C64Font PUTUDG  ok.
```

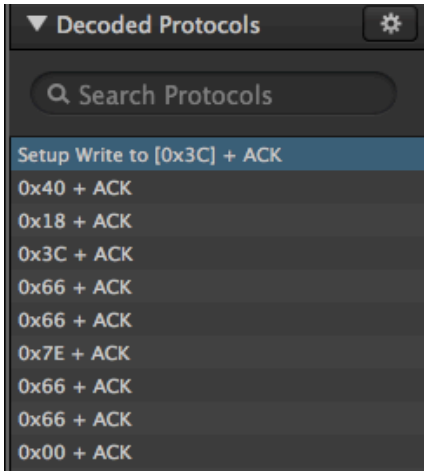



```
00011000
00111100
01100110
01100110
01111110
01100110
01100110
00000000
CHAR A FONT-SHAPE
```

Measured (Logic Analyzer) bus clock frequency is ~200kHz with 1/3rd duty cycle (1.625 µs for HIGH, 3.25 µs for LOW) as set by I2C-INIT

START and STOP conditions (SDA transition while SCL is high) are marked with a green and a red dot respectively





SSD1306 OLED Display Datasheet - (SSD1306 Datasheet for 096 OLED.pdf) page 33 - i2c-dip8.f SSD1306-dip-4.f fonts.f

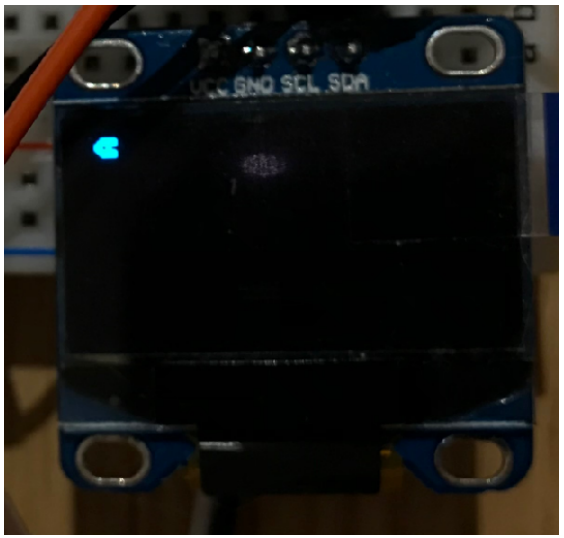
SSD1306 OLED DISPLAY

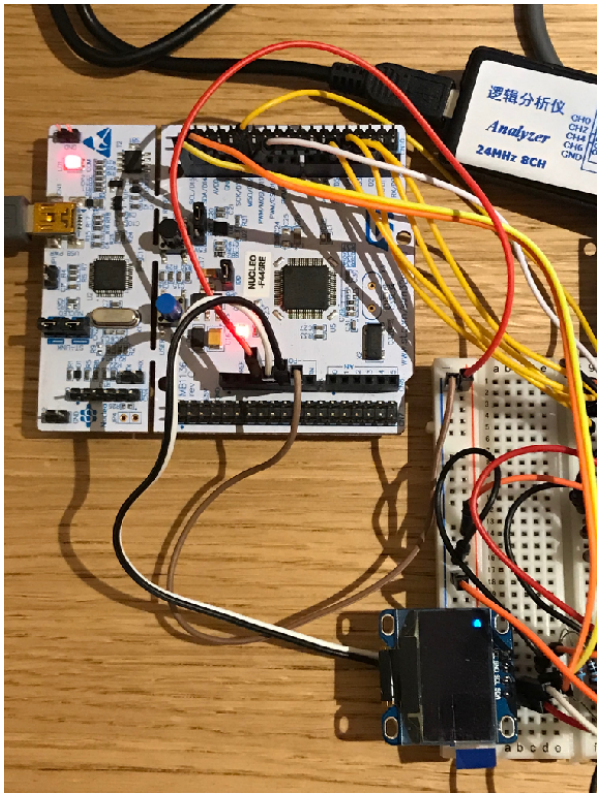
Displaying an 8x8 pixel character

- Due to the organization of the display memory, the character is displayed in the top left corner rotated by 90 degrees counterclockwise

```
CHAR A C64Font PUTUDG  ok.
```

```
00011000
00111100
01100110
01100110
01111110
01100110
01100110
00000000
CHAR A FONT-SHAPE
```





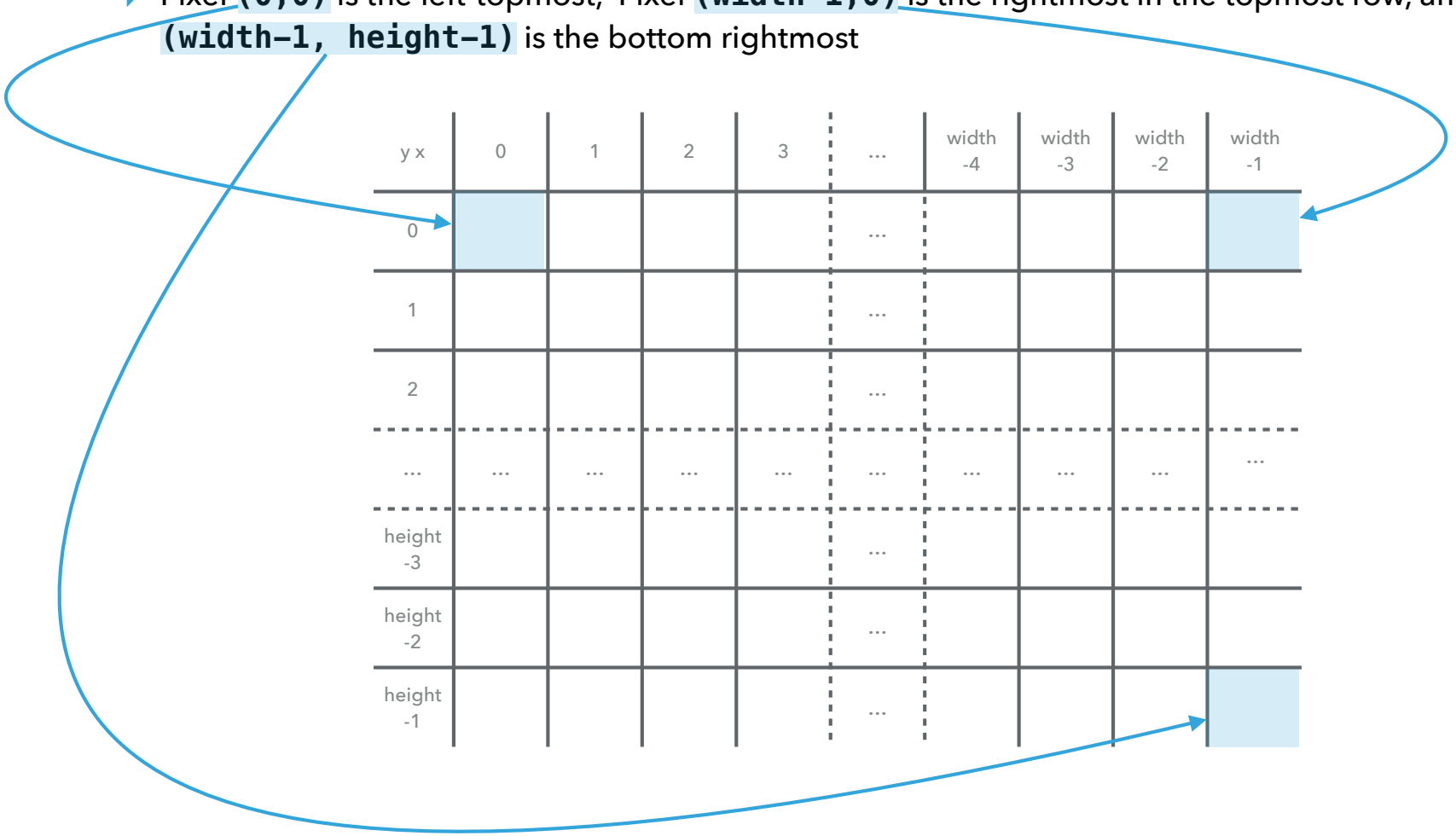
RPI 3B+ & 4B+ HDMI DISPLAY

- The Pi HDMI display is generated by the Videocore
- The display is a matrix of *width* * *height* pixels
  - Parameters *width* and *height* can be chosen and define the spatial resolution of the display

y x	0	1	2	3	...	width -4	width -3	width -2	width -1
0					...				
1					...				
2					...				
...	...	...	...	...	...	...	...	...	...
height -3					...				
height -2					...				
height -1					...				

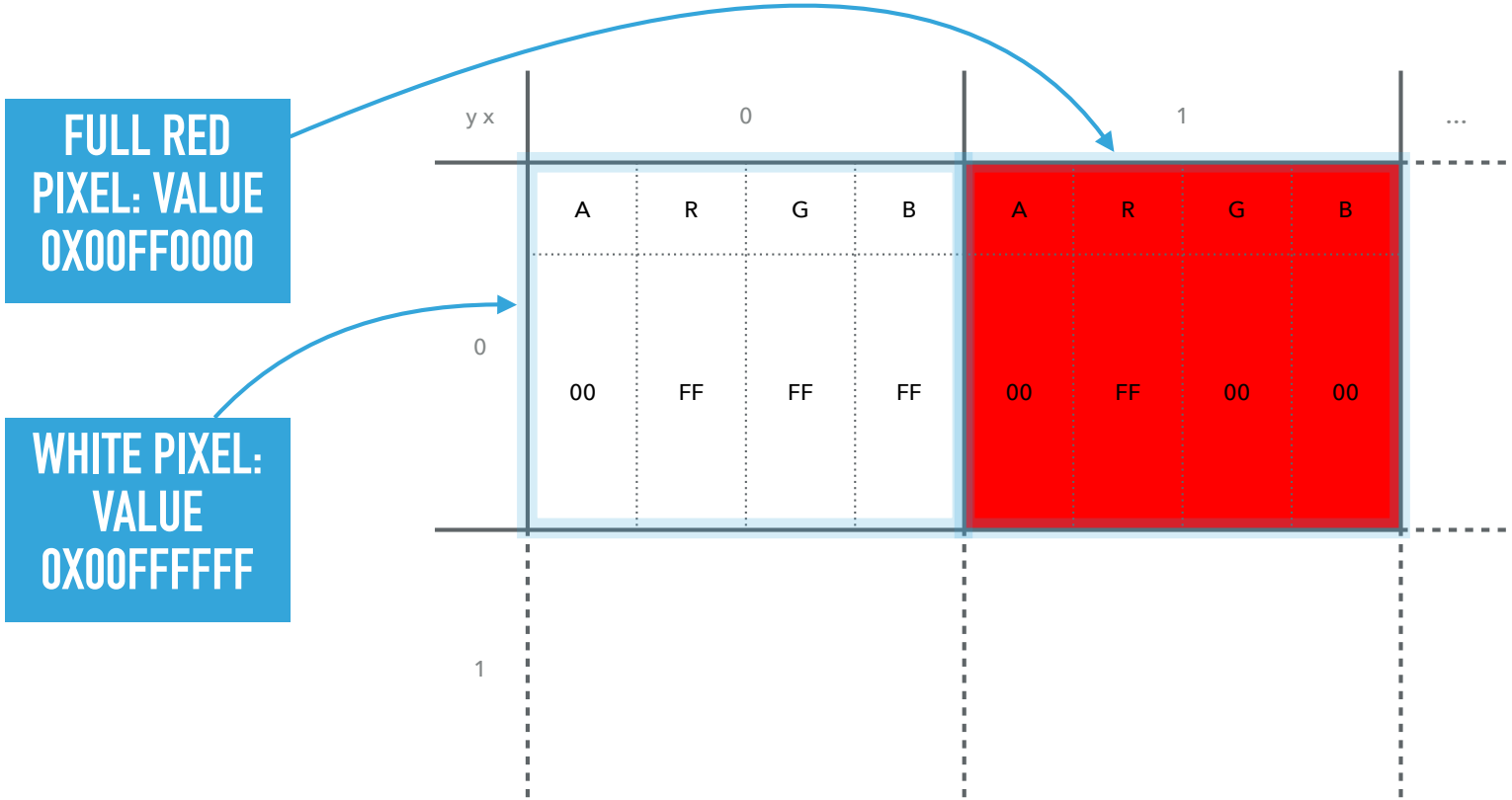
RPI 3B+ & 4B+ FRAMEBUFFER

- A pixel is located through its coordinates x={0,…,width-1} and y={0, …, height-1}
  - Pixel `(0,0)` is the left-topmost, Pixel `(width-1,0)` is the rightmost in the topmost row, and Pixel `(width-1, height-1)` is the bottom rightmost



RPI 3B+ & 4B+ FRAMEBUFFER

- The color of a pixel is controlled by a value encoded in *color_depth* bits (for the Pi *color_depth* can be any of 8, 16, and 32)
  - When using 32 bits per pixel (bpp) pixel color is encoded as an **ARGB** value: 8 bits for each of the **R**ed, **G**reen, and **B**lue color components, and 8 bits for the **A**lpha Channel used for transparency effects)



RPI 3B+ & 4B+ FRAMEBUFFER

- Pixel values are stored in a *framebuffer*, which is a memory region shared by the VC and the CPU
- The framebuffer is organized as a row-major matrix of *colordepth*-bit values
  - pixel values are packed one after another in a linear array of *colordepth*-bit values
    - the first item in the array controls the top leftmost pixel
    - in general, the color of pixel **`x,y`** is controlled by the i-th value in the linear array , where i=**`width*y+x`**

RPI 3B+ & 4B+ FRAMEBUFFER

- pijFORTHos-se configures the HDMI display for 1024x768x32bpp at startup
  - the color of pixel **`x,y`** is controlled by the ARGB value in the word at the byte offset **`4*(1024*y+x)`** from the framebuffer base address
  - the base address is shown in the output at startup

```
framebuffer at 0x3E8FA000 width 00000400 height 00000300 depth 00000020 pitch 00001000
0000480E
pijFORTHos 0.1.8-se-20190126 sp=0x00019090
```

RPI 3B+ & 4B+ FRAMEBUFFER

- Color of pixel (0,0) is changed to white:

```
HEX
00FFFFFF 3E8FA000 !
```

- To set the color of several contiguous pixels, an ARGB value and the address of the first pixel are left on the stack:

```
00FFFFFF 3E8FA000 1000 +
```

- the color is white and the address is that of pixel (0,1); then the phrase:

```
2DUP ! 4 +
```

- sets the color of the first pixel to white and leaves on the stack the color value and the address of the next pixel to the right: pixel (1,1)
- a horizontal line of **n** pixels is simply drawn by repeating the phrase **n** times

RPI 3B+ & 4B+ FRAMEBUFFER

- More conveniently a word is defined

```
: PR   2DUP !  4 + ;
```

- Similarly, to draw points downward the size in bytes of a line of pixels (4*1024=4*0x400=0x1000, the “pitch”) must be added to the address:

```
: PD   2DUP !  1000 + ;
```

- Warning: the address must be in the framebuffer range [base address, base address+(1024*768*4)-1]

RPI 3B+ & 4B+ FRAMEBUFFER

- Exercises:
  - Define a word to set the color of pixel (x,y)
    - Include framebuffer boundary checks
  - Define a set of words to draw lines in the eight discrete directions (north, northeast, east, southeast, south, southwest, west, northwest)
    - Include framebuffer boundary checks

RPI 3B+ & 4B+ DIGITAL INPUTS

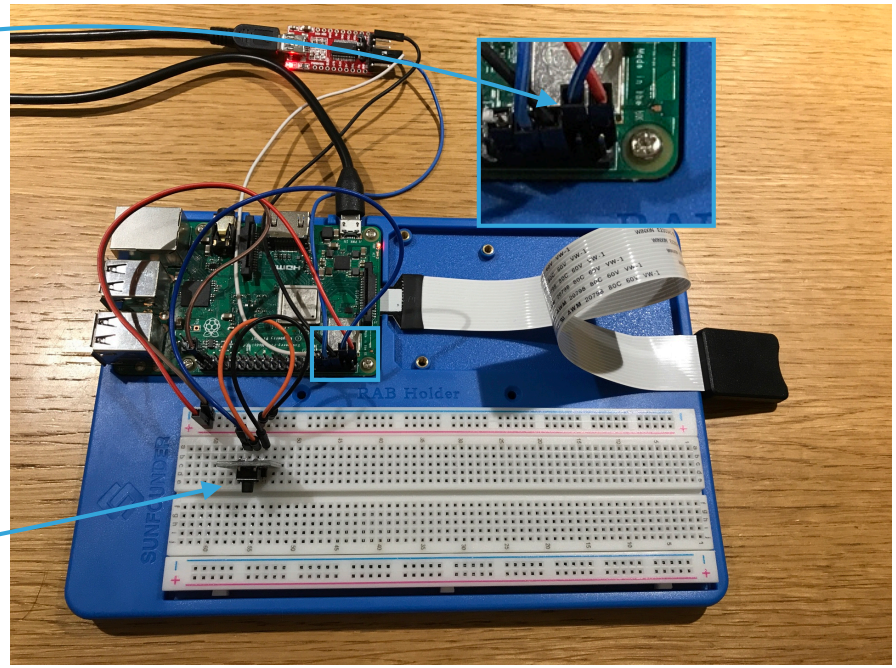
- General Purpose I/O (GPIO) GPLEVn registers are used to read levels on pins configured as digital inputs
  - The external circuit with a button provides pin 2 with
    - a logical 0 voltage level (GND) when pressed
    - a logical 1 voltage level (Vcc) when released

GPIO Pin Level Registers (GPLEVn)

SYNOPSIS The pin level registers return the actual value of the pin. The LEV{n} field gives the value of the respective GPIO pin.

Bit(s)	Field Name	Description	Type	Reset
31-0	LEVn (n=0..31)	0 = GPIO pin *n* is low 1 = GPIO pin *n* is high	R/W	0

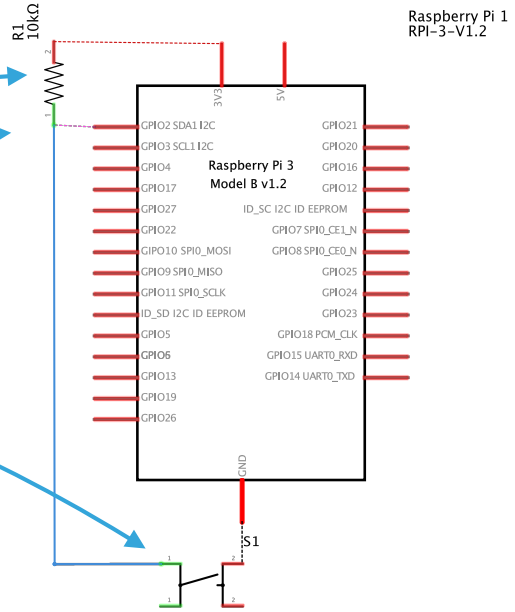
Table 6-12 – GPIO Level Register 0

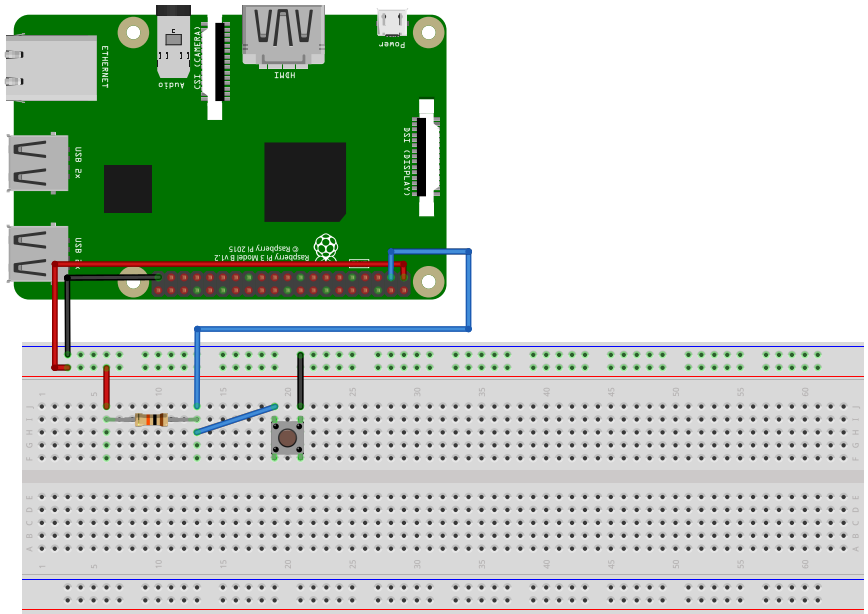


BCM2837-ARM-Peripherals.-.Revised.-.V2-1 **p. 96** - (BCM2837-ARM-Peripherals.-.Revised.-.V2-1.pdf)

RPI 3B+ & 4B+ DIGITAL INPUTS

- In this circuit a pull-up resistor ensures that pin 2 is provided with a stable connection to either Vcc or GND
  - When the button is released the resistor "pulls up" the voltage on pin 2 to the Vcc value so that a 1 is read
  - When the button is pressed pin 2 is "shorted" to GND and a 0 is read
  - If pin 2 were left "floating" it would be sensible to noise and erroneous values could possibly be read
- An inverted logic circuit would make use of a "pull-down" resistor connecting the pin to GND and the other contact of the button to Vcc
  - A button press would make the pin read a 1 while the released button would be read as a 0





RPI 3B+ & 4B+ DIGITAL INPUTS

- Internal pull-up or pull-down resistors can be activated for each GPIO Pin
- On the RPi 3B+ a procedure involving registers GPPUD and GPPUDCLKn is used to enable the same mode (pull-up, pull-down, no-pulls) for a set of pins at the same time
- On the RPi 4B+ the input mode of each pin is controlled by a 2-bit value in one of the GPIO_PUP_PDN_CNTRL_REGn registers

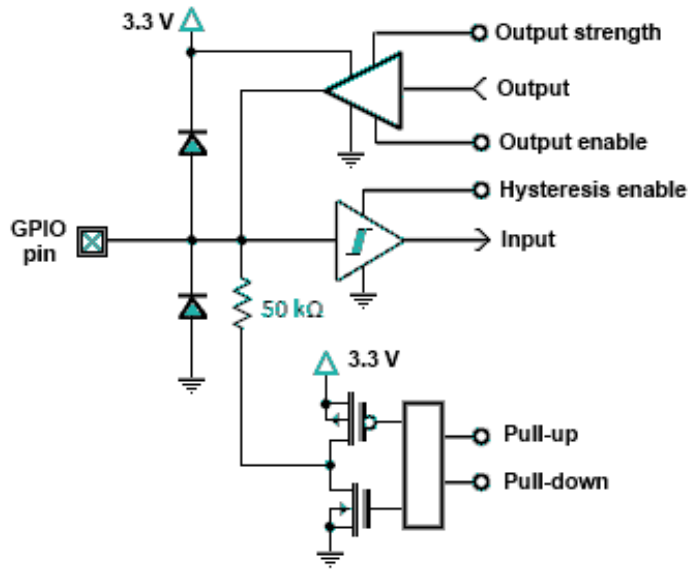
GPIO Pull-up/down Register (GPPUD)

SYNOPSIS

The GPIO Pull-up/down Register controls the actuation of the internal pull-up/down control line to ALL the GPIO pins. This register must be used in conjunction with the 2 GPPUDCLKn registers.

Note that it is not possible to read back the current Pull-up/down settings and so it is the users' responsibility to 'remember' which pull-up/downs are active. The reason for this is that GPIO pull-ups are maintained even in power-down mode when the core is off, when all register contents are lost.

The Alternate function table also has the pull state which is applied after a power down.



BCM2837-ARM-Peripherals.-.Revised.-.V2-1 **p. 100** - (BCM2837-ARM-Peripherals.-.Revised.-.V2-1.pdf) (Only RPi 3B+)

RPI 3B+ & 4B+ DIGITAL INPUTS

- The CPU can be relieved of much of the burden of waiting for the desired changes in digital inputs by programming the GPIO circuitry to detect an level change or edge events for each pin
- The GPEDSn registers record the detected events for each pin

GPIO Event Detect Status Registers (GPEDSn)

SYNOPSIS The event detect status registers are used to record level and edge events on the GPIO pins. The relevant bit in the event detect status registers is set whenever: 1) an edge is detected that matches the type of edge programmed in the rising/falling edge detect enable registers, or 2) a level is detected that matches the type of level programmed in the high/low level detect enable registers. The bit is cleared by writing a "1" to the relevant bit.

The interrupt controller can be programmed to interrupt the processor when any of the status bits are set. The GPIO peripheral has three dedicated interrupt lines. Each GPIO bank can generate an independent interrupt. The third line generates a single interrupt whenever any bit is set.

Bit(s)	Field Name	Description	Type	Reset
31-0	EDSn (n=0..31)	0 = Event not detected on GPIO pin n 1 = Event detected on GPIO pin n	R/W	0

Table 6-14 – GPIO Event Detect Status Register 0

Bit(s)	Field Name	Description	Type	Reset
31-22	-	Reserved	R	0
21-0	EDSn (n=32..53)	0 = Event not detected on GPIO pin n 1 = Event detected on GPIO pin n	R/W	0

Table 6-15 – GPIO Event Detect Status Register 1

RPI 3B+ & 4B+ DIGITAL INPUTS

- The GPIO circuitry can autonomously detect these events:
  - Rising edges (GPRENn)
  - Falling edges (GPFENn)
  - High level (GPHENn)
  - Low level (GPLENn)
  - Asynchronous rising edges (GPARENn)
  - Asynchronous falling edges (GPAFENn)

GPIO Rising Edge Detect Enable Registers (GPRENn)

SYNOPSIS The rising edge detect enable registers define the pins for which a rising edge transition sets a bit in the event detect status registers (GPEDSn). When the relevant bits are set in both the GPRENn and GPFENn registers, any transition (1 to 0 and 0 to 1) will set a bit in the GPEDSn registers. The GPRENn registers use synchronous edge detection. This means the input signal is sampled using the system clock and then it is looking for a "011" pattern on the sampled signal. This has the effect of suppressing glitches.

Bit(s)	Field Name	Description	Type	Reset
31-0	RENn (n=0..31)	0 = Rising edge detect disabled on GPIO pin *n*. 1 = Rising edge on GPIO pin *n* sets corresponding bit in EDSn.	R/W	0

Table 6-16 – GPIO Rising Edge Detect Status Register 0

Bit(s)	Field Name	Description	Type	Reset
31-22	-	Reserved	R	0
21-0	RENn (n=32..53)	0 = Rising edge detect disabled on GPIO pin *n*. 1 = Rising edge on GPIO pin *n* sets corresponding bit in EDSn.	R/W	0

Table 6-17 – GPIO Rising Edge Detect Status Register 1

BCM2837-ARM-Peripherals.-.Revised.-.V2-1 **p. 97** - (BCM2837-ARM-Peripherals.-.Revised.-.V2-1.pdf)

RPI 3B+ & 4B+ DIGITAL INPUTS

- The GPIO circuitry can autonomously detect these events:
  - Rising edges (GPRENn)
  - Falling edges (GPFENn)
  - High level (GPHENn)
  - Low level (GPLENn)
  - Asynchronous rising edges (GPARENn)
  - Asynchronous falling edges (GPAFENn)

GPIO Falling Edge Detect Enable Registers (GPRENn)

SYNOPSIS The falling edge detect enable registers define the pins for which a falling edge transition sets a bit in the event detect status registers (GPEDSn). When the relevant bits are set in both the GPRENn and GPFENn registers, any transition (1 to 0 and 0 to 1) will set a bit in the GPEDSn registers. The GPFENn registers use synchronous edge detection. This means the input signal is sampled using the system clock and then it is looking for a "100" pattern on the sampled signal. This has the effect of suppressing glitches.

Bit(s)	Field Name	Description	Type	Reset
31-0	FENn (n=0..31)	0 = Falling edge detect disabled on GPIO pin *n*. 1 = Falling edge on GPIO pin *n* sets corresponding bit in EDSn.	R/W	0

Table 6-18 – GPIO Falling Edge Detect Status Register 0

Bit(s)	Field Name	Description	Type	Reset
31-22	-	Reserved	R	0
21-0	FENn (n=32..53)	0 = Falling edge detect disabled on GPIO pin *n*. 1 = Falling edge on GPIO pin *n* sets corresponding bit in EDSn.	R/W	0

Table 6-19 – GPIO Falling Edge Detect Status Register 1

BCM2837-ARM-Peripherals.-.Revised.-.V2-1 **p. 98** - (BCM2837-ARM-Peripherals.-.Revised.-.V2-1.pdf)

RPI 3B+ & 4B+ DIGITAL INPUTS

- The GPIO circuitry can autonomously detect these events:
  - Rising edges (GPRENn)
  - Falling edges (GPFENn)
  - High level (GPHENn)
  - Low level (GPLENn)
  - Asynchronous rising edges (GPARENn)
  - Asynchronous falling edges (GPAFENn)

GPIO High Detect Enable Registers (GPHENn)				
SYNOPSIS	The high level detect enable registers define the pins for which a high level sets a bit in the event detect status register (GPEDSn). If the pin is still high when an attempt is made to clear the status bit in GPEDSn then the status bit will remain set.			
Bit(s)	**Field Name**	**Description**	**Type**	**Reset**
31-0	HENn (n=0..31)	0 = High detect disabled on GPIO pin *n* 1 = High on GPIO pin *n* sets corresponding bit in GPEDS	R/W	0

Table 6-20 – GPIO High Detect Status Register 0

Bit(s)	Field Name	Description	Type	Reset
31-22	-	Reserved	R	0
21-0	HENn (n=32..53)	0 = High detect disabled on GPIO pin *n* 1 = High on GPIO pin *n* sets corresponding bit in GPEDS	R/W	0

Table 6-21 – GPIO High Detect Status Register 1

BCM2837-ARM-Peripherals.-.Revised.-.V2-1 **p. 98** - (BCM2837-ARM-Peripherals.-.Revised.-.V2-1.pdf)

RPI 3B+ & 4B+ DIGITAL INPUTS

- The GPIO circuitry can autonomously detect these events:
  - Rising edges (GPRENn)
  - Falling edges (GPFENn)
  - High level (GPHENn)
  - Low level (GPLENn)
  - Asynchronous rising edges (GPARENn)
  - Asynchronous falling edges (GPAFENn)

GPIO Low Detect Enable Registers (GPLENn)

SYNOPSIS The low level detect enable registers define the pins for which a low level sets a bit in the event detect status register (GPEDSn). If the pin is still low when an attempt is made to clear the status bit in GPEDSn then the status bit will remain set.

Bit(s)	Field Name	Description	Type	Reset
31-0	LENn (n=0..31)	0 = Low detect disabled on GPIO pin *n* 1 = Low on GPIO pin *n* sets corresponding bit in GPEDS	R/W	0

Table 6-22 – GPIO Low Detect Status Register 0

Bit(s)	Field Name	Description	Type	Reset
31-22	-	Reserved	R	0
21-0	LENn (n=32..53)	0 = Low detect disabled on GPIO pin *n* 1 = Low on GPIO pin *n* sets corresponding bit in GPEDS	R/W	0

Table 6-23 – GPIO Low Detect Status Register 1

BCM2837-ARM-Peripherals.-.Revised.-.V2-1 **p. 99** - (BCM2837-ARM-Peripherals.-.Revised.-.V2-1.pdf)

RPI 3B+ & 4B+ DIGITAL INPUTS

- The GPIO circuitry can autonomously detect these events:
  - Rising edges (GPRENn)
  - Falling edges (GPFENn)
  - High level (GPHENn)
  - Low level (GPLENn)
  - Asynchronous rising edges (GPARENn)
  - Asynchronous falling edges (GPAFENn)

GPIO Asynchronous rising Edge Detect Enable Registers (GPARENn)

SYNOPSIS The asynchronous rising edge detect enable registers define the pins for which a asynchronous rising edge transition sets a bit in the event detect status registers (GPEDSn).

Asynchronous means the incoming signal is not sampled by the system clock. As such rising edges of very short duration can be detected.

Bit(s)	Field Name	Description	Type	Reset
31-0	ARENn (n=0..31)	0 = Asynchronous rising edge detect disabled on GPIO pin *n*. 1 = Asynchronous rising edge on GPIO pin *n* sets corresponding bit in EDSn.	R/W	0

Table 6-24 – GPIO Asynchronous rising Edge Detect Status Register 0

Bit(s)	Field Name	Description	Type	Reset
31-22	-	Reserved	R	0
21-0	ARENn (n=32..53)	0 = Asynchronous rising edge detect disabled on GPIO pin *n*. 1 = Asynchronous rising edge on GPIO pin *n* sets corresponding bit in EDSn.	R/W	0

BCM2837-ARM-Peripherals.-.Revised.-.V2-1 **p. 99** - (BCM2837-ARM-Peripherals.-.Revised.-.V2-1.pdf)

RPI 3B+ & 4B+ DIGITAL INPUTS

- The GPIO circuitry can autonomously detect these events:
  - Rising edges (GPRENn)
  - Falling edges (GPFENn)
  - High level (GPHENn)
  - Low level (GPLENn)
  - Asynchronous rising edges (GPARENn)
  - Asynchronous falling edges (GPAFENn)

GPIO Asynchronous Falling Edge Detect Enable Registers (GPAFENn)

SYNOPSIS The asynchronous falling edge detect enable registers define the pins for which a asynchronous falling edge transition sets a bit in the event detect status registers (GPEDSn). Asynchronous means the incoming signal is not sampled by the system clock. As such falling edges of very short duration can be detected.

Bit(s)	Field Name	Description	Type	Reset
31-0	AFENn (n=0..31)	0 = Asynchronous falling edge detect disabled on GPIO pin *n*. 1 = Asynchronous falling edge on GPIO pin *n* sets corresponding bit in EDSn.	R/W	0

Table 6-26 – GPIO Asynchronous Falling Edge Detect Status Register 0

Bit(s)	Field Name	Description	Type	Reset
31-22	-	Reserved	R	0
21-0	AFENn (n=32..53)	0 = Asynchronous falling edge detect disabled on GPIO pin *n*. 1 = Asynchronous falling edge on GPIO pin *n* sets corresponding bit in EDSn.	R/W	0

Table 6-27 – GPIO Asynchronous Falling Edge Detect Status Register 1

BCM2837-ARM-Peripherals.-.Revised.-.V2-1 **p. 100** - (BCM2837-ARM-Peripherals.-.Revised.-.V2-1.pdf)