Chapter 14
ARM GPIO
(General Purpose Input/Output)

Embedded Systems with ARM Cortex-M
Updated 3/6/2018
GPIO Pins

• Flexibility due to limited amount of pins on the board.

• Can be configured as 4 types:
  - Digital INPUT
  - Digital OUTPUT
  - Analog INPUT (ADC)/OUTPUT (DAC)
  - Alternate Function (USART)
I/O Electronics

Figure 16. Basic structure of a five-volt tolerant I/O port bit

1. $V_{DD_{FT}}$ is a potential specific to five-volt tolerant I/Os and different from $V_{DD}$. 
Digital OUTPUT

Figure 19. Output configuration
Digital INPUT

Figure 18. Input floating/pull up/pull down configurations

Read
Write
Read/write

Input data register
Input data register

TTL Schmitt trigger
input driver
output driver

I/O pin

protection diode
protection diode

V_{DD}
V_{SS}
V_{DD}
V_{SS}

on/off
on/off
pull up
pull down
Analog IN/OUT

Figure 21. High impedance-analog configuration
Alternate Function

Figure 20. Alternate function configuration
8.4.4 GPIO port pull-up/pull-down register (GPIOx_PUPDR) (x = A..E and H)

Address offset: 0x0C

Reset values:
- 0x6400 0000 for port A
- 0x0000 0100 for port B
- 0x0000 0000 for other ports

<table>
<thead>
<tr>
<th>31</th>
<th>30</th>
<th>29</th>
<th>28</th>
<th>27</th>
<th>26</th>
<th>25</th>
<th>24</th>
<th>23</th>
<th>22</th>
<th>21</th>
<th>20</th>
<th>19</th>
<th>18</th>
<th>17</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
</tr>
<tr>
<td>15</td>
<td>14</td>
<td>13</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
</tr>
</tbody>
</table>

Bits 2y:2y+1 PUPDRy[1:0]: Port x configuration bits (y = 0..15)
These bits are written by software to configure the I/O pull-up or pull-down
00: No pull-up, pull-down
01: Pull-up
10: Pull-down
11: Reserved

8.4.5 GPIO port input data register (GPIOx_IDR) (x = A..E and H)

Address offset: 0x10

Reset value: 0x0000 XXXX (where X means undefined)

<table>
<thead>
<tr>
<th>31</th>
<th>30</th>
<th>29</th>
<th>28</th>
<th>27</th>
<th>26</th>
<th>25</th>
<th>24</th>
<th>23</th>
<th>22</th>
<th>21</th>
<th>20</th>
<th>19</th>
<th>18</th>
<th>17</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>14</td>
<td>13</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>IDR15</td>
<td>IDR14</td>
<td>IDR13</td>
<td>IDR12</td>
<td>IDR11</td>
<td>IDR10</td>
<td>IDR9</td>
<td>IDR8</td>
<td>IDR7</td>
<td>IDR6</td>
<td>IDR5</td>
<td>IDR4</td>
<td>IDR3</td>
<td>IDR2</td>
<td>IDR1</td>
<td>IDR0</td>
</tr>
<tr>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
</tr>
</tbody>
</table>

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 IDRy: Port input data (y = 0..15)
These bits are read-only and can be accessed in word mode only. They contain the input value of the corresponding I/O port.
All pins are configured in SW!

• By default all peripherals are not clocked. You must enable the clock to each peripheral you wish to use.

• You must then READ the datasheet and find out which registers need to be changed to get desired configuration.
Memory Map of STM32F401
Peripheral Registers

2.3 Memory map

See the datasheet corresponding to your device for a comprehensive diagram of the memory map. *Table 1* gives the boundary addresses of the peripherals available in STM32F411xC/E device.

<table>
<thead>
<tr>
<th>Boundary address</th>
<th>Peripheral</th>
<th>Bus</th>
<th>Register map</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x5000 0000 - 0x5003 FFFF</td>
<td>USB OTG FS</td>
<td>AHB2</td>
<td><strong>Section 22.16.6: OTG_FS register map on page 744</strong></td>
</tr>
<tr>
<td>0x4002 6400 - 0x4002 67FF</td>
<td>DMA2</td>
<td></td>
<td><strong>Section 9.5.11: DMA register map on page 194</strong></td>
</tr>
<tr>
<td>0x4002 6000 - 0x4002 63FF</td>
<td>DMA1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x4002 3C00 - 0x4002 3FFF</td>
<td>Flash interface register</td>
<td></td>
<td><strong>Section 3.8: Flash interface registers on page 58</strong></td>
</tr>
<tr>
<td>0x4002 3800 - 0x4002 3BFF</td>
<td>RCC</td>
<td></td>
<td><strong>Section 6.3.22: RCC register map on page 133</strong></td>
</tr>
<tr>
<td>0x4002 3000 - 0x4002 33FF</td>
<td>CRC</td>
<td>AHB1</td>
<td><strong>Section 4.4.4: CRC register map on page 68</strong></td>
</tr>
<tr>
<td>0x4002 1C00 - 0x4002 1FFF</td>
<td>GPIOH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x4002 1000 - 0x4002 13FF</td>
<td>GPIOE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x4002 0C00 - 0x4002 0FF</td>
<td>GPIOD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x4002 0800 - 0x4002 0BFF</td>
<td>GPIOC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x4002 0400 - 0x4002 07FF</td>
<td>GPIOB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x4002 0000 - 0x4002 03FF</td>
<td>GPIOA</td>
<td></td>
<td><strong>Section 8.4.11: GPIO register map on page 160</strong></td>
</tr>
</tbody>
</table>
Figure 12. Clock tree
### 6.3.9 RCC AHB1 peripheral clock enable register (RCC_AHB1ENR)

**Address offset:** 0x30  
**Reset value:** 0x0000 0000  
**Access:** no wait state, word, half-word and byte access.

<table>
<thead>
<tr>
<th>31</th>
<th>30</th>
<th>29</th>
<th>28</th>
<th>27</th>
<th>26</th>
<th>25</th>
<th>24</th>
<th>23</th>
<th>22</th>
<th>21</th>
<th>20</th>
<th>19</th>
<th>18</th>
<th>17</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DMA2EN</td>
<td>DMA1EN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>14</td>
<td>13</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Reserved</td>
<td>CRCEN</td>
<td>Reserved</td>
<td>GPIOHEN</td>
<td>Reserved</td>
<td>GPIOEEN</td>
<td>GPIOEN</td>
<td>GPIOEN</td>
<td>GPIOEN</td>
<td>GPIOEN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Bits 31:23** Reserved, must be kept at reset value.
- **Bit 22 DMA2EN**: DMA2 clock enable  
  - Set and cleared by software.  
  - 0: DMA2 clock disabled  
  - 1: DMA2 clock enabled
- **Bit 21 DMA1EN**: DMA1 clock enable  
  - Set and cleared by software.  
  - 0: DMA1 clock disabled  
  - 1: DMA1 clock enabled
- **Bits 20:13** Reserved, must be kept at reset value.
- **Bit 12 CRCEN**: CRC clock enable  
  - Set and cleared by software.  
  - 0: CRC clock disabled  
  - 1: CRC clock enabled
- **Bits 11:8** Reserved, must be kept at reset value.
- **Bit 7 GPIOHEN**: IO port H clock enable  
  - Set and reset by software.
Steps to set up Digital OUTPUT

• Provide CLOCK to the port
• Configure the desired I/O as OUTPUT or INPUT in the GPIOx_MODER register
• View the RESET values for the other registers that control I/O and change if necessary:
  - GPIOx_OTYPER – Open-Drain/Push-Pull
  - GPIOx_OSPEEDR – Low/Med/High Speed
  - GPIOx_PUPDR – (Weak) P/U, P/D, None
Calculating Delay Loop

• Desired delay time = 500ms
• Clock Frequency = 10MHz (Prove it)
• $10 \times 10^6$ (cycles/second) $\times$ 500 $\times$ $10^{-3}$ seconds = 40000000 cycles
• Instructions in loop?
Board Schematic