

Raspberry Peripheral Driver

Generated by Doxygen 1.8.9.1

Wed Aug 19 2015 01:07:35

Contents

1	Class Index	1
1.1	Class List	1
2	File Index	1
2.1	File List	2
3	Class Documentation	2
3.1	PwmRPi Class Reference	2
3.1.1	Detailed Description	2
3.1.2	Constructor & Destructor Documentation	2
3.1.3	Member Function Documentation	2
3.1.4	Member Data Documentation	3
4	File Documentation	3
4.1	Arduino.h File Reference	3
4.1.1	Macro Definition Documentation	3
4.2	Arduino.h	3
4.3	Pwm.cpp File Reference	4
4.3.1	Variable Documentation	4
4.4	Pwm.cpp	4
4.5	Pwm.h File Reference	4
4.5.1	Macro Definition Documentation	6
4.5.2	Variable Documentation	9
4.6	Pwm.h	9
4.7	simple_read.c File Reference	10
4.7.1	Function Documentation	10
4.8	simple_read.c	10
4.9	using_external_eeprom.c File Reference	11
4.9.1	Function Documentation	11
4.10	using_external_eeprom.c	11
	Index	13

1 Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

PwmRPi	2
-------------------------------	----------

2 File Index

2.1 File List

Here is a list of all files with brief descriptions:

Arduino.h	3
Pwm.cpp	4
Pwm.h	4
simple_read.c	10
using_external_eeprom.c	11

3 Class Documentation

3.1 PwmRPi Class Reference

```
#include <Pwm.h>
```

Public Member Functions

- [PwmRPi](#) (unsigned char [channel](#))
- void [begin](#) ()
- void [stop](#) ()
- void [analogWrite](#) (unsigned char value)

Private Attributes

- Bcm2835::Peripheral [pwm](#)
- unsigned char [channel](#)

3.1.1 Detailed Description

Definition at line 73 of file [Pwm.h](#).

3.1.2 Constructor & Destructor Documentation

3.1.2.1 PwmRPi::PwmRPi (unsigned char *channel*)

Definition at line 4 of file [Pwm.cpp](#).

3.1.3 Member Function Documentation

3.1.3.1 void PwmRPi::analogWrite (unsigned char *value*)

3.1.3.2 void PwmRPi::begin ()

Definition at line 8 of file [Pwm.cpp](#).

3.1.3.3 void PwmRPi::stop ()

Definition at line 13 of file [Pwm.cpp](#).

3.1.4 Member Data Documentation

3.1.4.1 unsigned char PwmRPi::channel [private]

PWM pin.

Definition at line 83 of file [Pwm.h](#).

3.1.4.2 Bcm2835::Peripheral PwmRPi::pwm [private]

Peripheral.

Definition at line 78 of file [Pwm.h](#).

The documentation for this class was generated from the following files:

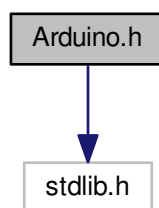
- [Pwm.h](#)
- [Pwm.cpp](#)

4 File Documentation

4.1 Arduino.h File Reference

```
#include <stdlib.h>
```

Include dependency graph for Arduino.h:



Macros

- `#define delay(n) usleep(1000 * n)`

4.1.1 Macro Definition Documentation

4.1.1.1 `#define delay(n) usleep(1000 * n)`

Definition at line 3 of file [Arduino.h](#).

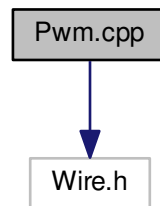
4.2 Arduino.h

```
00001 #include <stdlib.h>
00002
00003 #define delay(n) usleep(1000 * n)
```

4.3 Pwm.cpp File Reference

```
#include "Wire.h"
```

Include dependency graph for Pwm.cpp:



Variables

- [PwmRPi Pwm0](#) (0)
- [PwmRPi Pwm1](#) (1)

4.3.1 Variable Documentation

4.3.1.1 PwmRPi Pwm0(0)

4.3.1.2 PwmRPi Pwm1(1)

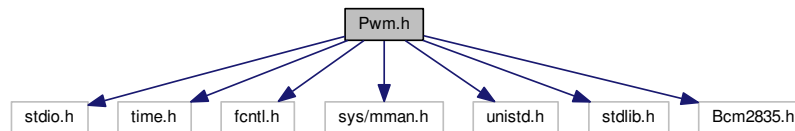
4.4 Pwm.cpp

```
00001
00002 #include "Wire.h"
00003
00004 PwmRPi::PwmRPi(unsigned char channel) {
00005     this->channel = (channel & 0x01);
00006 }
00007
00008 void PwmRPi::begin() {
00009     pwm.address = PWM_ADDRESS;
00010     Bcm2835::mapPeripheral(&pwm);
00011 }
00012
00013 void PwmRPi::stop() {
00014     Bcm2835::unmapPeripheral(&pwm);
00015 }
00016
00017 PwmRPi Pwm0(0);
00018 PwmRPi Pwm1(1);
```

4.5 Pwm.h File Reference

```
#include <stdio.h>
#include <time.h>
#include <fcntl.h>
#include <sys/mman.h>
#include <unistd.h>
#include <stdlib.h>
#include <Bcm2835.h>
```

Include dependency graph for Pwm.h:



Classes

- class [PwmRPi](#)

Macros

- #define [PWM_ADDRESS](#) 0x20c000
- #define [PWM_CTL](#) *((unsigned int *) (pwm.mem) + 0x00)
- #define [PWM_STA](#) *((unsigned int *) (pwm.mem) + 0x01)
- #define [PWM_DMAC](#) *((unsigned int *) (pwm.mem) + 0x02)
- #define [PWM_RNG1](#) *((unsigned int *) (pwm.mem) + 0x03)
- #define [PWM_DAT1](#) *((unsigned int *) (pwm.mem) + 0x04)
- #define [PWM_DAT1](#) *((unsigned int *) (pwm.mem) + 0x05)
- #define [PWM_FIF1](#) *((unsigned int *) (pwm.mem) + 0x06)
- #define [PWM_RNG2](#) *((unsigned int *) (pwm.mem) + 0x07)
- #define [PWM_DAT2](#) *((unsigned int *) (pwm.mem) + 0x08)
- #define [PWM_CTL_PWEN1](#) (0x01 << 0)
- #define [PWM_CTL_MODE1](#) (0x01 << 1)
- #define [PWM_CTL_RPTL1](#) (0x01 << 2)
- #define [PWM_CTL_SBIT1](#) (0x01 << 3)
- #define [PWM_CTL_POLA1](#) (0x01 << 4)
- #define [PWM_CTL_USEF1](#) (0x01 << 5)
- #define [PWM_CTL_CLRF1](#) (0x01 << 6)
- #define [PWM_CTL_MSEN1](#) (0x01 << 7)
- #define [PWM_CTL_PWEN2](#) (0x01 << 8)
- #define [PWM_CTL_MODE2](#) (0x01 << 9)
- #define [PWM_CTL_RPTL2](#) (0x01 << 10)
- #define [PWM_CTL_SBIT2](#) (0x01 << 11)
- #define [PWM_CTL_POLA2](#) (0x01 << 12)
- #define [PWM_CTL_USEF2](#) (0x01 << 13)
- #define [PWM_CTL_MSEN2](#) (0x01 << 15)
- #define [PWM_STA_FULL1](#) (0x01 << 0)
- #define [PWM_STA_EMPTY1](#) (0x01 << 1)
- #define [PWM_STA_WERR1](#) (0x01 << 2)

- `#define PWM_STA_RERR1 (0x01 << 3)`
- `#define PWM_STA_GAPO1 (0x01 << 4)`
- `#define PWM_STA_GAPO2 (0x01 << 5)`
- `#define PWM_STA_GAPO3 (0x01 << 6)`
- `#define PWM_STA_GAPO4 (0x01 << 7)`
- `#define PWM_STA_BERR (0x01 << 8)`
- `#define PWM_STA_STA1 (0x01 << 9)`
- `#define PWM_STA_STA2 (0x01 << 10)`
- `#define PWM_STA_STA3 (0x01 << 11)`
- `#define PWM_STA_STA4 (0x01 << 12)`
- `#define PWM_DMAC_DREQ (0xff << 0)`
- `#define PWM_DMAC_PANIC (0xff << 8)`
- `#define PWM_DMAC_ENAB (0x01 << 31)`

Variables

- `PwmRPi Pwm0`
- `PwmRPi Pwm1`

4.5.1 Macro Definition Documentation

4.5.1.1 `#define PWM_ADDRESS 0x20c000`

This is a simple Wire library to Raspberry.

It doesn't use the specific i2c module (`i2c_dev` or `i2c_bcm2708`) it maps the memory (the BSC0 chunk) into the virtual memory space and handles directly the register.

Thanks to this blog: <http://www.susa.net/wordpress/2012/06/raspberry-pi-pcf8563-real-time-clock>

Definition at line 23 of file `Pwm.h`.

4.5.1.2 `#define PWM_CTL *((unsigned int *) (pwm.mem) + 0x00)`

Definition at line 25 of file `Pwm.h`.

4.5.1.3 `#define PWM_CTL_CLRF1 (0x01 << 6)`

Definition at line 42 of file `Pwm.h`.

4.5.1.4 `#define PWM_CTL_MODE1 (0x01 << 1)`

Definition at line 37 of file `Pwm.h`.

4.5.1.5 `#define PWM_CTL_MODE2 (0x01 << 9)`

Definition at line 46 of file `Pwm.h`.

4.5.1.6 `#define PWM_CTL_MSEN1 (0x01 << 7)`

Definition at line 43 of file `Pwm.h`.

4.5.1.7 `#define PWM_CTL_MSEN2 (0x01 << 15)`

Definition at line 51 of file `Pwm.h`.

4.5.1.8 `#define PWM_CTL_POLA1 (0x01 << 4)`

Definition at line 40 of file `Pwm.h`.

4.5.1.9 `#define PWM_CTL_POLA2 (0x01 << 12)`

Definition at line 49 of file [Pwm.h](#).

4.5.1.10 `#define PWM_CTL_PWEN1 (0x01 << 0)`

Definition at line 36 of file [Pwm.h](#).

4.5.1.11 `#define PWM_CTL_PWEN2 (0x01 << 8)`

Definition at line 45 of file [Pwm.h](#).

4.5.1.12 `#define PWM_CTL_RPTL1 (0x01 << 2)`

Definition at line 38 of file [Pwm.h](#).

4.5.1.13 `#define PWM_CTL_RPTL2 (0x01 << 10)`

Definition at line 47 of file [Pwm.h](#).

4.5.1.14 `#define PWM_CTL_SBIT1 (0x01 << 3)`

Definition at line 39 of file [Pwm.h](#).

4.5.1.15 `#define PWM_CTL_SBIT2 (0x01 << 11)`

Definition at line 48 of file [Pwm.h](#).

4.5.1.16 `#define PWM_CTL_USEF1 (0x01 << 5)`

Definition at line 41 of file [Pwm.h](#).

4.5.1.17 `#define PWM_CTL_USEF2 (0x01 << 13)`

Definition at line 50 of file [Pwm.h](#).

4.5.1.18 `#define PWM_DAT1 *((unsigned int *) (pwm.mem) + 0x04)`

Definition at line 30 of file [Pwm.h](#).

4.5.1.19 `#define PWM_DAT1 *((unsigned int *) (pwm.mem) + 0x05)`

Definition at line 30 of file [Pwm.h](#).

4.5.1.20 `#define PWM_DAT2 *((unsigned int *) (pwm.mem) + 0x08)`

Definition at line 33 of file [Pwm.h](#).

4.5.1.21 `#define PWM_DMAC *((unsigned int *) (pwm.mem) + 0x02)`

Definition at line 27 of file [Pwm.h](#).

4.5.1.22 `#define PWM_DMAC_DREQ (0xff << 0)`

Definition at line 69 of file [Pwm.h](#).

4.5.1.23 `#define PWM_DMAC_ENAB (0x01 << 31)`

Definition at line 71 of file [Pwm.h](#).

4.5.1.24 `#define PWM_DMAC_PANIC (0xff << 8)`

Definition at line 70 of file [Pwm.h](#).

4.5.1.25 `#define PWM_FIF1 *((unsigned int *) (pwm.mem) + 0x06)`

Definition at line 31 of file [Pwm.h](#).

4.5.1.26 `#define PWM_RNG1 *((unsigned int *) (pwm.mem) + 0x03)`

Definition at line 28 of file [Pwm.h](#).

4.5.1.27 `#define PWM_RNG2 *((unsigned int *) (pwm.mem) + 0x07)`

Definition at line 32 of file [Pwm.h](#).

4.5.1.28 `#define PWM_STA *((unsigned int *) (pwm.mem) + 0x01)`

Definition at line 26 of file [Pwm.h](#).

4.5.1.29 `#define PWM_STA_BERR (0x01 << 8)`

Definition at line 62 of file [Pwm.h](#).

4.5.1.30 `#define PWM_STA_EMPT1 (0x01 << 1)`

Definition at line 55 of file [Pwm.h](#).

4.5.1.31 `#define PWM_STA_FULL1 (0x01 << 0)`

Definition at line 54 of file [Pwm.h](#).

4.5.1.32 `#define PWM_STA_GAPO1 (0x01 << 4)`

Definition at line 58 of file [Pwm.h](#).

4.5.1.33 `#define PWM_STA_GAPO2 (0x01 << 5)`

Definition at line 59 of file [Pwm.h](#).

4.5.1.34 `#define PWM_STA_GAPO3 (0x01 << 6)`

Definition at line 60 of file [Pwm.h](#).

4.5.1.35 `#define PWM_STA_GAPO4 (0x01 << 7)`

Definition at line 61 of file [Pwm.h](#).

4.5.1.36 `#define PWM_STA_RERR1 (0x01 << 3)`

Definition at line 57 of file [Pwm.h](#).

4.5.1.37 `#define PWM_STA_STA1 (0x01 << 9)`

Definition at line 63 of file [Pwm.h](#).

4.5.1.38 `#define PWM_STA_STA2 (0x01 << 10)`

Definition at line 64 of file [Pwm.h](#).

4.5.1.39 #define PWM_STA_STA3 (0x01 << 11)

Definition at line 65 of file [Pwm.h](#).

4.5.1.40 #define PWM_STA_STA4 (0x01 << 12)

Definition at line 66 of file [Pwm.h](#).

4.5.1.41 #define PWM_STA_WERR1 (0x01 << 2)

Definition at line 56 of file [Pwm.h](#).

4.5.2 Variable Documentation

4.5.2.1 PwmRpi Pwm0

4.5.2.2 PwmRpi Pwm1

4.6 Pwm.h

```

00001
00011 #ifndef __RASPBERRY_WIRE_H__
00012 #define __RASPBERRY_WIRE_H__ 1
00013
00014 #include <stdio.h>
00015 #include <time.h>
00016 #include <fcntl.h>
00017 #include <sys/mman.h>
00018 #include <unistd.h>
00019 #include <stdlib.h>
00020
00021 #include <Bcm2835.h>
00022
00023 #define PWM_ADDRESS      0x20c000
00024
00025 #define PWM_CTL          *((unsigned int *) (pwm.mem) + 0x00)
00026 #define PWM_STA          *((unsigned int *) (pwm.mem) + 0x01)
00027 #define PWM_DMACH        *((unsigned int *) (pwm.mem) + 0x02)
00028 #define PWM_RNG1         *((unsigned int *) (pwm.mem) + 0x03)
00029 #define PWM_DAT1         *((unsigned int *) (pwm.mem) + 0x04)
00030 #define PWM_DAT1         *((unsigned int *) (pwm.mem) + 0x05)
00031 #define PWM_FIFO         *((unsigned int *) (pwm.mem) + 0x06)
00032 #define PWM_RNG2         *((unsigned int *) (pwm.mem) + 0x07)
00033 #define PWM_DAT2         *((unsigned int *) (pwm.mem) + 0x08)
00034
00035 // p142
00036 #define PWM_CTL_PWEN1    (0x01 << 0)
00037 #define PWM_CTL_MODE1    (0x01 << 1)
00038 #define PWM_CTL_RPTL1    (0x01 << 2)
00039 #define PWM_CTL_SBIT1    (0x01 << 3)
00040 #define PWM_CTL_POLA1    (0x01 << 4)
00041 #define PWM_CTL_USEF1    (0x01 << 5)
00042 #define PWM_CTL_CLRF1    (0x01 << 6)
00043 #define PWM_CTL_MSEN1    (0x01 << 7)
00044
00045 #define PWM_CTL_PWEN2    (0x01 << 8)
00046 #define PWM_CTL_MODE2    (0x01 << 9)
00047 #define PWM_CTL_RPTL2    (0x01 << 10)
00048 #define PWM_CTL_SBIT2    (0x01 << 11)
00049 #define PWM_CTL_POLA2    (0x01 << 12)
00050 #define PWM_CTL_USEF2    (0x01 << 13)
00051 #define PWM_CTL_MSEN2    (0x01 << 15)
00052
00053 // p144
00054 #define PWM_STA_FULLL1    (0x01 << 0)
00055 #define PWM_STA_EMPTY1    (0x01 << 1)
00056 #define PWM_STA_WERR1    (0x01 << 2)
00057 #define PWM_STA_RERR1    (0x01 << 3)
00058 #define PWM_STA_GAPO1     (0x01 << 4)
00059 #define PWM_STA_GAPO2     (0x01 << 5)
00060 #define PWM_STA_GAPO3     (0x01 << 6)
00061 #define PWM_STA_GAPO4     (0x01 << 7)
00062 #define PWM_STA_BERR       (0x01 << 8)
00063 #define PWM_STA_STA1       (0x01 << 9)
00064 #define PWM_STA_STA2       (0x01 << 10)
00065 #define PWM_STA_STA3       (0x01 << 11)
00066 #define PWM_STA_STA4       (0x01 << 12)
00067

```

```

00068 // p145
00069 #define PWM_DMAC_DREQ (0xff << 0)
00070 #define PWM_DMAC_PANIC (0xff << 8)
00071 #define PWM_DMAC_ENAB (0x01 << 31)
00072
00073 class PwmRPi {
00074     Bcm2835::Peripheral pwm;
00079     unsigned char channel;
00084 public:
00086     PwmRPi(unsigned char channel);
00088     void begin();
00092     void stop();
00095     void analogWrite(unsigned char value);
00099
00100 };
00101
00102 extern PwmRPi Pwm0;
00103 extern PwmRPi Pwm1;
00104
00105 #endif /* __RASPBERRY_PWM_H__ */

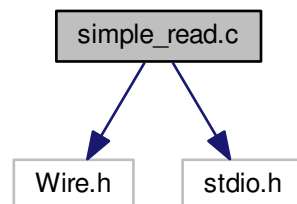
```

4.7 simple_read.c File Reference

```
#include <Wire.h>
```

```
#include <stdio.h>
```

Include dependency graph for simple_read.c:



Functions

- int [main](#) ()

4.7.1 Function Documentation

4.7.1.1 int main ()

Definition at line 4 of file [simple_read.c](#).

4.8 simple_read.c

```

00001 #include <Wire.h>
00002 #include <stdio.h>
00003

```

```

00004 int main() {
00005     printf("Begining...");
00006     Wire.begin();
00007
00008     Wire.beginTransaction(0x21);
00009     Wire.write(0x00);
00010     Wire.write(0x00);
00011     Wire.endTransmission();
00012
00013     Wire.requestFrom(0x21, 16);
00014     while (Wire.available()) {
00015         printf("%d\n", Wire.read());
00016     }
00017     Wire.dumpStatus();
00018     Wire.stop();
00019     printf("Stop.");
00020 }

```

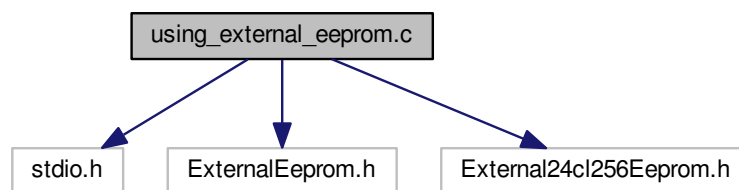
4.9 using_external_eeprom.c File Reference

```

#include <stdio.h>
#include <ExternalEeprom.h>
#include <External24cl256Eeprom.h>

```

Include dependency graph for using_external_eeprom.c:



Functions

- int `main` ()

4.9.1 Function Documentation

4.9.1.1 int main ()

Definition at line 8 of file [using_external_eeprom.c](#).

4.10 using_external_eeprom.c

```

00001
00002 // Yout will find ExternalEeprom here: https://github.com/dalmirdasilva/ArduinoMemoryDriver
00003
00004 #include <stdio.h>
00005 #include <ExternalEeprom.h>
00006 #include <External24cl256Eeprom.h>
00007
00008 int main() {
00009     int i;
00010     External24cl256Eeprom eeprom(0x00);
00011
00012     for (i = 0; i < 16; i++) {
00013         printf("%d\n", eeprom.read(i));
00014     }
00015 }

```

```
00016     return 0;  
00017 }  
00018  
00019
```

Index

analogWrite
 PwmRPi, [2](#)
Arduino.h, [3](#)
 delay, [3](#)

begin
 PwmRPi, [2](#)

channel
 PwmRPi, [3](#)

delay
 Arduino.h, [3](#)

main
 simple_read.c, [10](#)
 using_external_eeprom.c, [11](#)

PWM_ADDRESS
 Pwm.h, [6](#)
PWM_CTL
 Pwm.h, [6](#)
PWM_CTL_CLRF1
 Pwm.h, [6](#)
PWM_CTL_MODE1
 Pwm.h, [6](#)
PWM_CTL_MODE2
 Pwm.h, [6](#)
PWM_CTL_MSEN1
 Pwm.h, [6](#)
PWM_CTL_MSEN2
 Pwm.h, [6](#)
PWM_CTL_POLA1
 Pwm.h, [6](#)
PWM_CTL_POLA2
 Pwm.h, [6](#)
PWM_CTL_PWEN1
 Pwm.h, [6](#)
PWM_CTL_PWEN2
 Pwm.h, [6](#)
PWM_CTL_RPTL1
 Pwm.h, [7](#)
PWM_CTL_RPTL2
 Pwm.h, [7](#)
PWM_CTL_SBIT1
 Pwm.h, [7](#)
PWM_CTL_SBIT2
 Pwm.h, [7](#)
PWM_CTL_USEF1
 Pwm.h, [7](#)
PWM_CTL_USEF2
 Pwm.h, [7](#)
PWM_DAT1
 Pwm.h, [7](#)
PWM_DAT2
 Pwm.h, [7](#)

PWM_DMACH
 Pwm.h, [7](#)
PWM_DMACH_DREQ
 Pwm.h, [7](#)
PWM_DMACH_ENAB
 Pwm.h, [7](#)
PWM_DMACH_PANIC
 Pwm.h, [7](#)
PWM_FIF1
 Pwm.h, [7](#)
PWM_RNG1
 Pwm.h, [7](#)
PWM_RNG2
 Pwm.h, [8](#)
PWM_STA
 Pwm.h, [8](#)
PWM_STA_BERR
 Pwm.h, [8](#)
PWM_STA_EMPT1
 Pwm.h, [8](#)
PWM_STA_FULL1
 Pwm.h, [8](#)
PWM_STA_GAPO1
 Pwm.h, [8](#)
PWM_STA_GAPO2
 Pwm.h, [8](#)
PWM_STA_GAPO3
 Pwm.h, [8](#)
PWM_STA_GAPO4
 Pwm.h, [8](#)
PWM_STA_RERR1
 Pwm.h, [8](#)
PWM_STA_STA1
 Pwm.h, [8](#)
PWM_STA_STA2
 Pwm.h, [8](#)
PWM_STA_STA3
 Pwm.h, [8](#)
PWM_STA_STA4
 Pwm.h, [8](#)
PWM_STA_WERR1
 Pwm.h, [8](#)
pwm
 PwmRPi, [3](#)
Pwm.cpp, [4](#)
 Pwm0, [4](#)
 Pwm1, [4](#)
Pwm.h, [4](#)
 PWM_ADDRESS, [6](#)
 PWM_CTL, [6](#)
 PWM_CTL_CLRF1, [6](#)
 PWM_CTL_MODE1, [6](#)
 PWM_CTL_MODE2, [6](#)
 PWM_CTL_MSEN1, [6](#)
 PWM_CTL_MSEN2, [6](#)

- PWM_CTL_POLA1, [6](#)
- PWM_CTL_POLA2, [6](#)
- PWM_CTL_PWEN1, [6](#)
- PWM_CTL_PWEN2, [6](#)
- PWM_CTL_RPTL1, [7](#)
- PWM_CTL_RPTL2, [7](#)
- PWM_CTL_SBIT1, [7](#)
- PWM_CTL_SBIT2, [7](#)
- PWM_CTL_USEF1, [7](#)
- PWM_CTL_USEF2, [7](#)
- PWM_DAT1, [7](#)
- PWM_DAT2, [7](#)
- PWM_DMAC, [7](#)
- PWM_DMAC_DREQ, [7](#)
- PWM_DMAC_ENAB, [7](#)
- PWM_DMAC_PANIC, [7](#)
- PWM_FIF1, [7](#)
- PWM_RNG1, [7](#)
- PWM_RNG2, [8](#)
- PWM_STA, [8](#)
- PWM_STA_BERR, [8](#)
- PWM_STA_EMPTY1, [8](#)
- PWM_STA_FULL1, [8](#)
- PWM_STA_GAPO1, [8](#)
- PWM_STA_GAPO2, [8](#)
- PWM_STA_GAPO3, [8](#)
- PWM_STA_GAPO4, [8](#)
- PWM_STA_RERR1, [8](#)
- PWM_STA_STA1, [8](#)
- PWM_STA_STA2, [8](#)
- PWM_STA_STA3, [8](#)
- PWM_STA_STA4, [8](#)
- PWM_STA_WERR1, [8](#)
- Pwm0, [9](#)
- Pwm1, [9](#)
- Pwm0
 - Pwm.cpp, [4](#)
 - Pwm.h, [9](#)
- Pwm1
 - Pwm.cpp, [4](#)
 - Pwm.h, [9](#)
- PwmRPi, [2](#)
 - analogWrite, [2](#)
 - begin, [2](#)
 - channel, [3](#)
 - pwm, [3](#)
 - PwmRPi, [2](#)
 - stop, [2](#)
- simple_read.c, [10](#)
 - main, [10](#)
- stop
 - PwmRPi, [2](#)
- using_external_eeprom.c, [11](#)
 - main, [11](#)