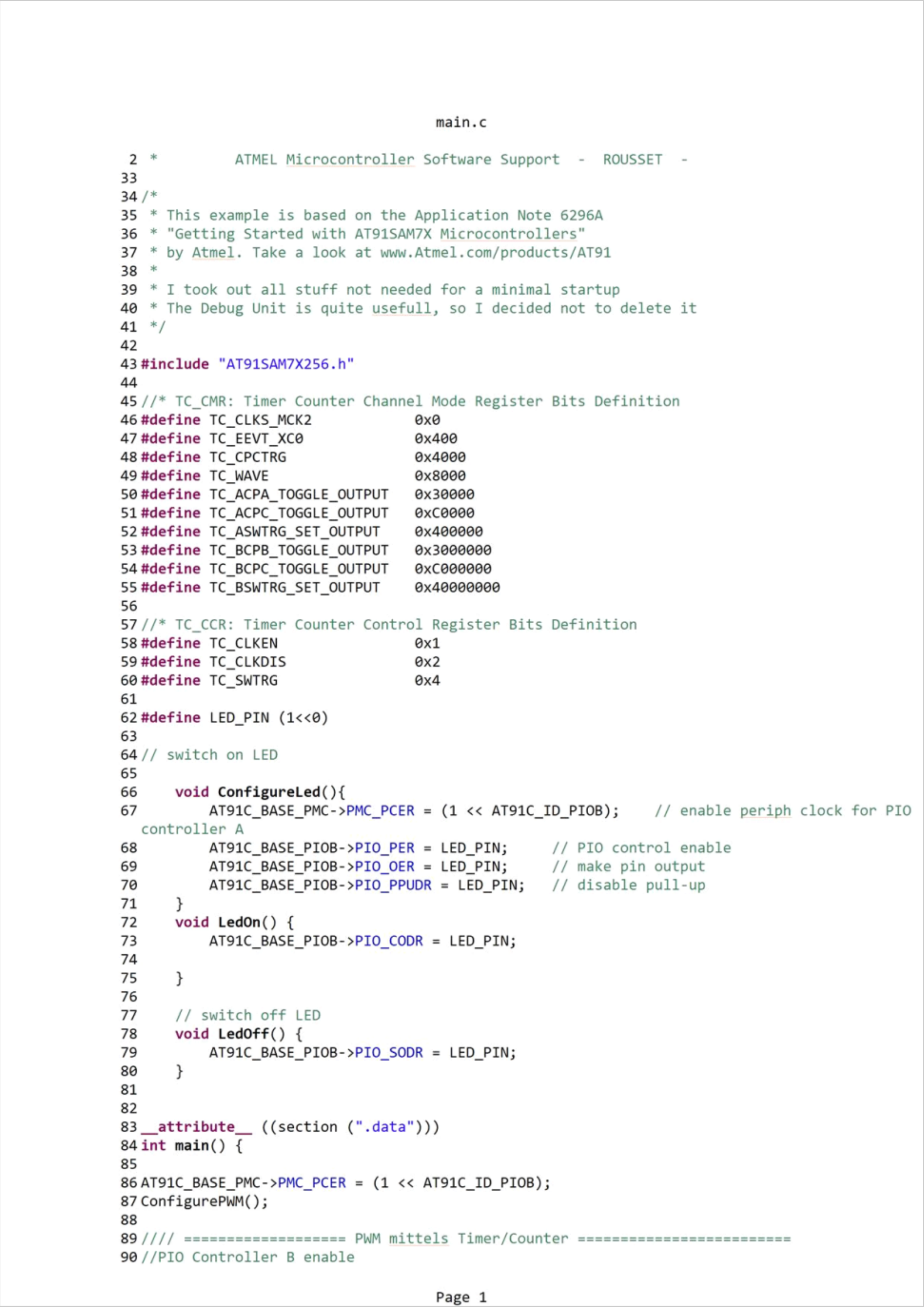
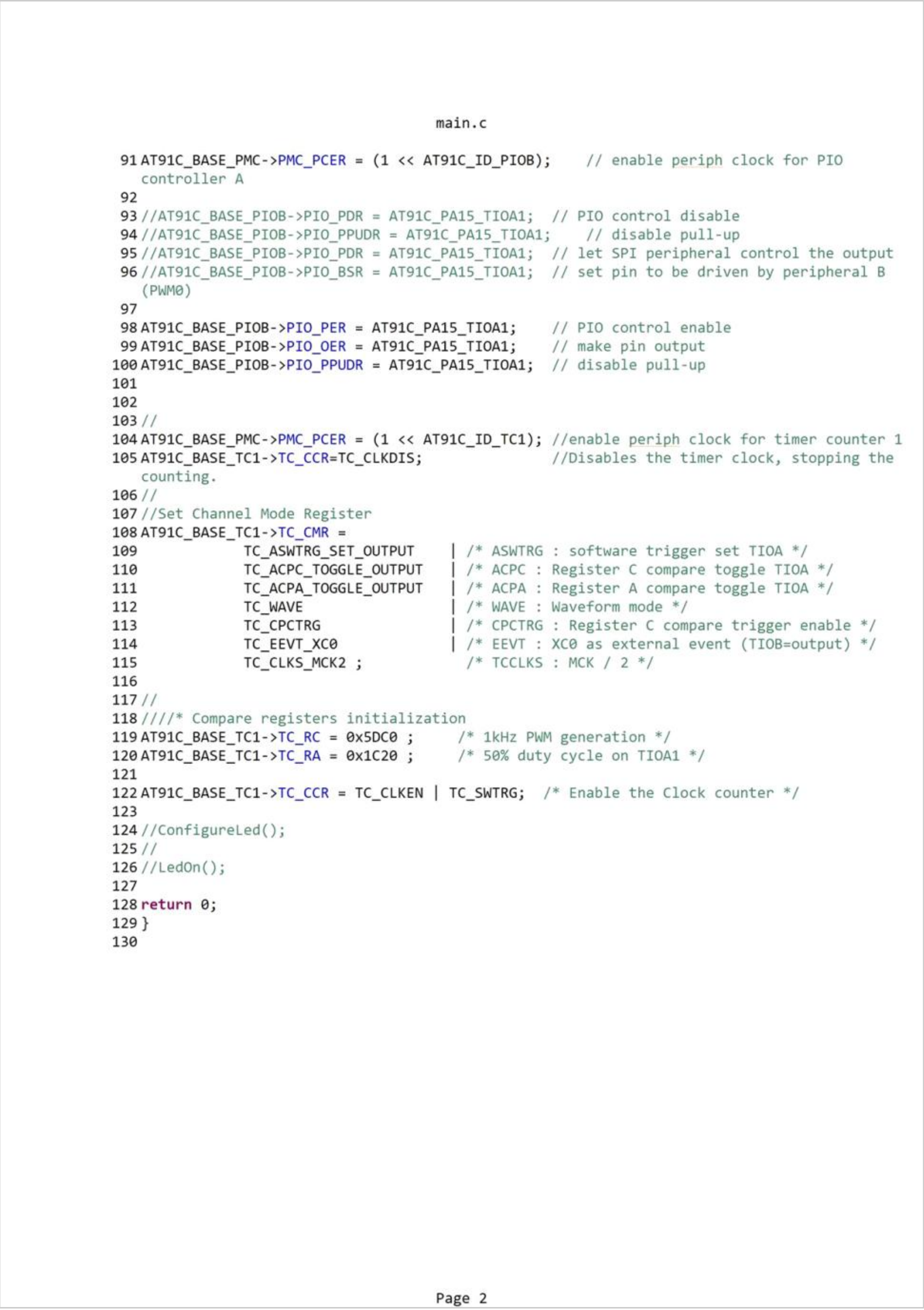


main.c

```
 *         ATMEL Microcontroller Software Support  -  ROUSSET  -

/*
 * This example is based on the Application Note 6296A
 * "Getting Started with AT91SAM7X Microcontrollers"
 * by Atmel. Take a look at www.Atmel.com/products/AT91
 *
 * I took out all stuff not needed for a minimal startup
 * The Debug Unit is quite usefull, so I decided not to delete it
 */

#include "AT91SAM7X256.h"

//* TC_CMR: Timer Counter Channel Mode Register Bits Definition
#define TC_CLKS_MCK2            0x0
#define TC_EEVT_XC0             0x400
#define TC_CPCTRG               0x4000
#define TC_WAVE                 0x8000
#define TC_ACPA_TOGGLE_OUTPUT   0x30000
#define TC_ACPC_TOGGLE_OUTPUT   0xC0000
#define TC_ASWTRG_SET_OUTPUT    0x400000
#define TC_BCPB_TOGGLE_OUTPUT   0x3000000
#define TC_BCPC_TOGGLE_OUTPUT   0xC000000
#define TC_BSWTRG_SET_OUTPUT    0x40000000

//* TC_CCR: Timer Counter Control Register Bits Definition
#define TC_CLKEN                0x1
#define TC_CLKDIS               0x2
#define TC_SWTRG                0x4

#define LED_PIN (1<<0)

// switch on LED

    void ConfigureLed(){
        AT91C_BASE_PMC->PMC_PCER = (1 << AT91C_ID_PIOB);    // enable periph clock for PIO controller A
        AT91C_BASE_PIOB->PIO_PER = LED_PIN;     // PIO control enable
        AT91C_BASE_PIOB->PIO_OER = LED_PIN;     // make pin output
        AT91C_BASE_PIOB->PIO_PPUDR = LED_PIN;   // disable pull-up
    }
    void LedOn() {
        AT91C_BASE_PIOB->PIO_CODR = LED_PIN;

    }

    // switch off LED
    void LedOff() {
        AT91C_BASE_PIOB->PIO_SODR = LED_PIN;
    }


__attribute__ ((section (".data")))
int main() {

AT91C_BASE_PMC->PMC_PCER = (1 << AT91C_ID_PIOB);
ConfigurePWM();

//// =================== PWM mittels Timer/Counter =========================
//PIO Controller B enable
```

Page 1



main.c

```
AT91C_BASE_PMC->PMC_PCER = (1 << AT91C_ID_PIOB);    // enable periph clock for PIO controller A

//AT91C_BASE_PIOB->PIO_PDR = AT91C_PA15_TIOA1;  // PIO control disable
//AT91C_BASE_PIOB->PIO_PPUDR = AT91C_PA15_TIOA1;    // disable pull-up
//AT91C_BASE_PIOB->PIO_PDR = AT91C_PA15_TIOA1;  // let SPI peripheral control the output
//AT91C_BASE_PIOB->PIO_BSR = AT91C_PA15_TIOA1;  // set pin to be driven by peripheral B (PWM0)

AT91C_BASE_PIOB->PIO_PER = AT91C_PA15_TIOA1;    // PIO control enable
AT91C_BASE_PIOB->PIO_OER = AT91C_PA15_TIOA1;    // make pin output
AT91C_BASE_PIOB->PIO_PPUDR = AT91C_PA15_TIOA1;  // disable pull-up


//
AT91C_BASE_PMC->PMC_PCER = (1 << AT91C_ID_TC1); //enable periph clock for timer counter 1
AT91C_BASE_TC1->TC_CCR=TC_CLKDIS;               //Disables the timer clock, stopping the counting.
//
//Set Channel Mode Register
AT91C_BASE_TC1->TC_CMR =
            TC_ASWTRG_SET_OUTPUT    | /* ASWTRG : software trigger set TIOA */
            TC_ACPC_TOGGLE_OUTPUT   | /* ACPC : Register C compare toggle TIOA */
            TC_ACPA_TOGGLE_OUTPUT   | /* ACPA : Register A compare toggle TIOA */
            TC_WAVE                 | /* WAVE : Waveform mode */
            TC_CPCTRG               | /* CPCTRG : Register C compare trigger enable */
            TC_EEVT_XC0             | /* EEVT : XC0 as external event (TIOB=output) */
            TC_CLKS_MCK2 ;            /* TCCLKS : MCK / 2 */

//
////* Compare registers initialization
AT91C_BASE_TC1->TC_RC = 0x5DC0 ;     /* 1kHz PWM generation */
AT91C_BASE_TC1->TC_RA = 0x1C20 ;     /* 50% duty cycle on TIOA1 */

AT91C_BASE_TC1->TC_CCR = TC_CLKEN | TC_SWTRG;  /* Enable the Clock counter */

//ConfigureLed();
//
//LedOn();

return 0;
}

```