Reference Design

Motion Sensing & LED Control with a ZMOTION/ IXYS Display

RD000302-0814

Abstract

This reference design demonstrates how to use the Z8FS040 ZMOTION MCU with the IXYS LDS8710 High-Efficiency LED Driver and explains how to implement features such as ambient light detection and LED dimming in addition to passive infrared motion detection. It also illustrates how the combined capabilities of both products are an exceptional choice for energy management functions in applications ranging from low-power displays and backlighting to ambient room lighting control.

Note: The source code files associated with this reference design, RD0003-SC01.zip, RD0003-SC02.zip and RD0003-SC03.zip, are available for download on zilog.com. These files have been tested with version 5.0.0 of ZDS II for Z8 Encore! XP MCUs. Subsequent releases of ZDS II may require you to modify the code supplied with this reference design.

Features

This Motion Sensing and LED Control reference design features:

- Integrated passive infrared (PIR) motion sensing and LED control combined in a single 8-pin Zilog MCU
- Self-contained, single-board solution for motion sensing and LED control
- Motion detection adapts to various lenses with simple header file changes
- Selectable power source of either two AA batteries or an external power supply with operating voltage in the 2.7V to 3.6V range
- Ability to switch high-intensity LEDs on and off in response to motion
- Controls LED brightness from three sources:
  - Console commands
  - Ambient light
  - Potentiometer (control source depends on software build)
- Available in handheld size, approximately 2" x 3"

Discussion

This Motion Sensing and LED Control reference design consists of a Zilog Z8FS040 ZMOTION MCU and an IXYS LDS8710 High-Efficiency LED Driver. LED brightness is controlled by a PWM output from the Z8FS040 MCU, with inputs from either a potentiometer, ambient light sensor (ALS), or console via the UART interface (collectively referred to as *control switches* in this document). Individual project files are provided for each of these three control switch options.

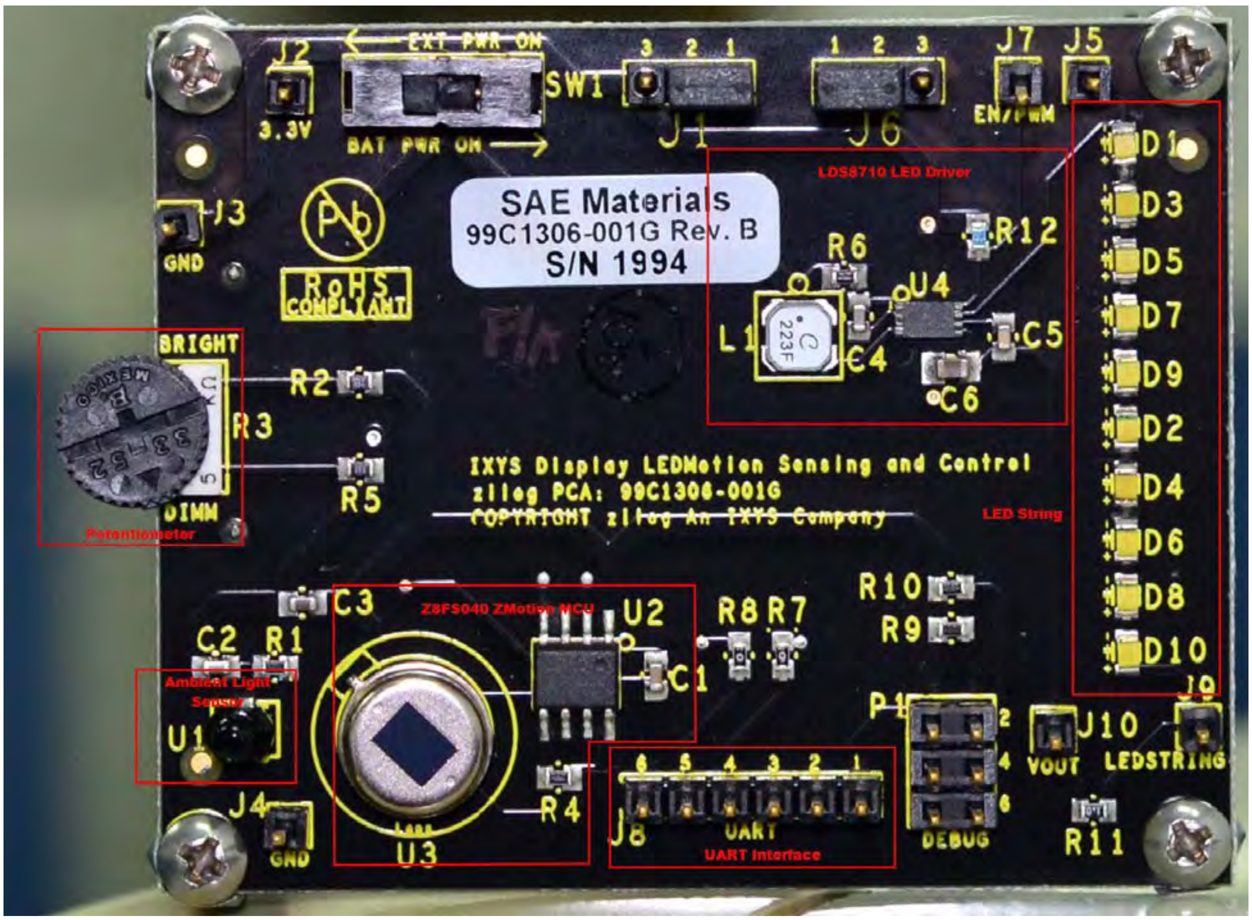

Figure 1. Motion Sensing and LED Control Board

Z8FS040 ZMOTION MCU

The Z8FS040 ZMOTION MCU is part of Zilog's ZMOTION Detection and Control Family, which provides an integrated and flexible solution for motion detection applications based on passive infrared (PIR) technology. The family includes the Z8FS040 microcontroller with integrated motion detection algorithms, plus a selection of lenses and pyroelectric sensors to fit a wide range of application requirements. Optimized configuration parameters for each lens are provided to ensure the best possible performance while significantly reducing development risk and minimizing time to market. Zilog's PIR motion detection technology provides a dramatic improvement in both sensitivity and stability over traditional designs, and is scalable to many market segments including lighting control, HVAC, access control, vending, display, proximity, power management, occupancy sensing and many others.

The key features of the ZMOTION Z8FS040 MCU are:

- High performance eZ8® MCU core
- Integrated software-based motion detection algorithms controlled and monitored through API registers
- 4KB in-circuit programmable Flash available for application code
- API settings provided for each lens and pyroelectric sensor combination
- Directly supports 1 or 2 pyroelectric sensors
- Dynamically configurable sensitivity, range and frequency response parameters
- Directional detection mode
- Internal precision oscillator running at 5.53MHz or external oscillator operating up to 20MHz
- High-resolution 4-channel Sigma/Delta ADC on an 8-pin SOIC package (PIR input uses 1 channel)
- On-chip analog comparator with independent programmable reference voltage
- Full-duplex UART with dedicated BRG
- Two 16-bit timers with input capture, output compare and PWM capabilities (11 modes total)
- Single-pin debug with unlimited breakpoints
- 2.7V to 3.6V operating voltage with extended operating temperature range (–40°C to +105°C)
- Low-power modes

LDS8710 High-Efficiency LED Driver

The IXYS LDS8710 High-Efficiency LED Driver is a fixed-frequency current-mode boost converter with an internal synchronous rectifier and a cycle-by-cycle switch current limit specifically designed to drive a string of up to 10 white LEDs. Series connection of these LEDs provides constant current and uniform brightness, eliminating the requirement for ballast resistors and factory calibration.

The main features of IXYS' LDS8710 High-Efficiency LED Driver are:

- High-efficiency boost converter with input voltage in the 2.7V to 5.5V range
- No external Schottky required (an internal synchronous rectifier is used instead)

- 250mV current sense voltage
- Drives one LED string with 10 LEDs in series up to 30mA
- 0.7MHz switching frequency
- Efficiency greater than 83%
- PWM LED dimming control mode
- Overvoltage, undervoltage, overcurrent and overtemperature protection
- Low shutdown current (<1µA)
- Available in an 8-pin 2" x 3" x 0.8mm TDFN package

Hardware Architecture

The ZMOTION Z8FS040 MCU acts as the main controller for this reference design. The block diagram in Figure 2 shows a basic architecture for this reference design, using one of three control sources – ambient light sensor, UART interface or potentiometer – to control the LDS8710 driver via a Pulse-Width Modulator (PWM). The LDS8710 driver uses the duty cycle of the PWM to control the current (and, therefore, the brightness) of the LED string. The Z8FS040 MCU also directly interfaces to a pyroelectric sensor for its motion detection functions. For more details about these hardware connections, see Schematics on page 17.

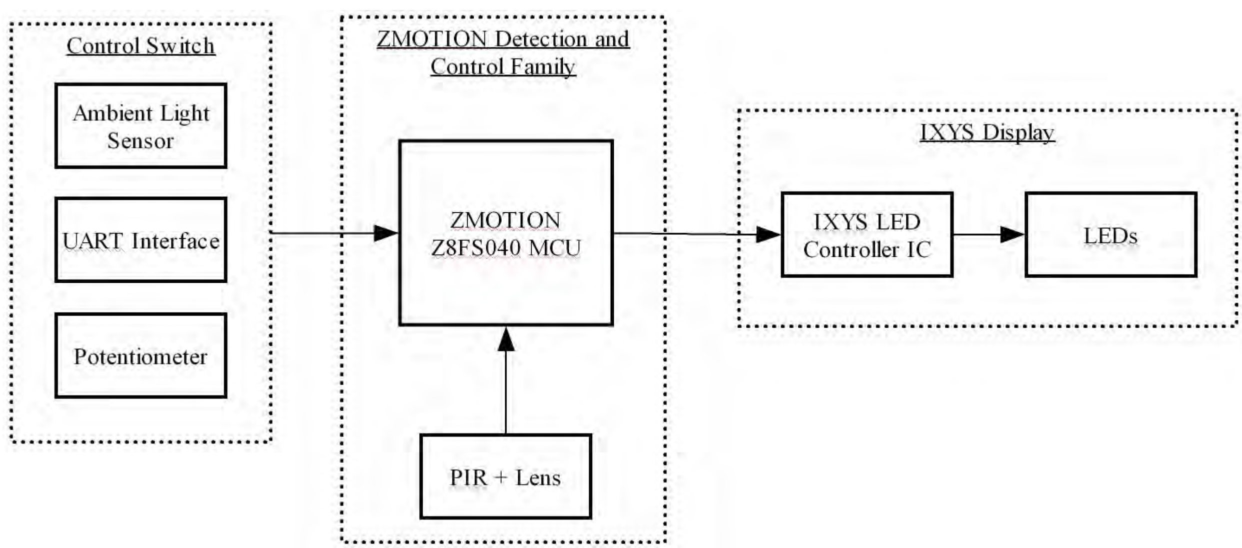

Figure 2. Motion Sensing and LED Control Block Diagram

Z8FS040 ZMOTION MCU Circuit Description

Figure 3 shows the debug and pyroelectric sensor interface for the ZMOTION MCU. The interface to the pyroelectric sensor is via the dedicated input ANA2 (pin 5); only a 47K drain resistor is required. The RST and DBG (pins 2 and 4, respectively) are used for the debug interface, and can be optionally used for other functions; 10K pull-up resistors are required for each of these signals. Pins 3, 6 and 7 are free to be used for other application requirements; these pins are used for the control switches in this reference design.

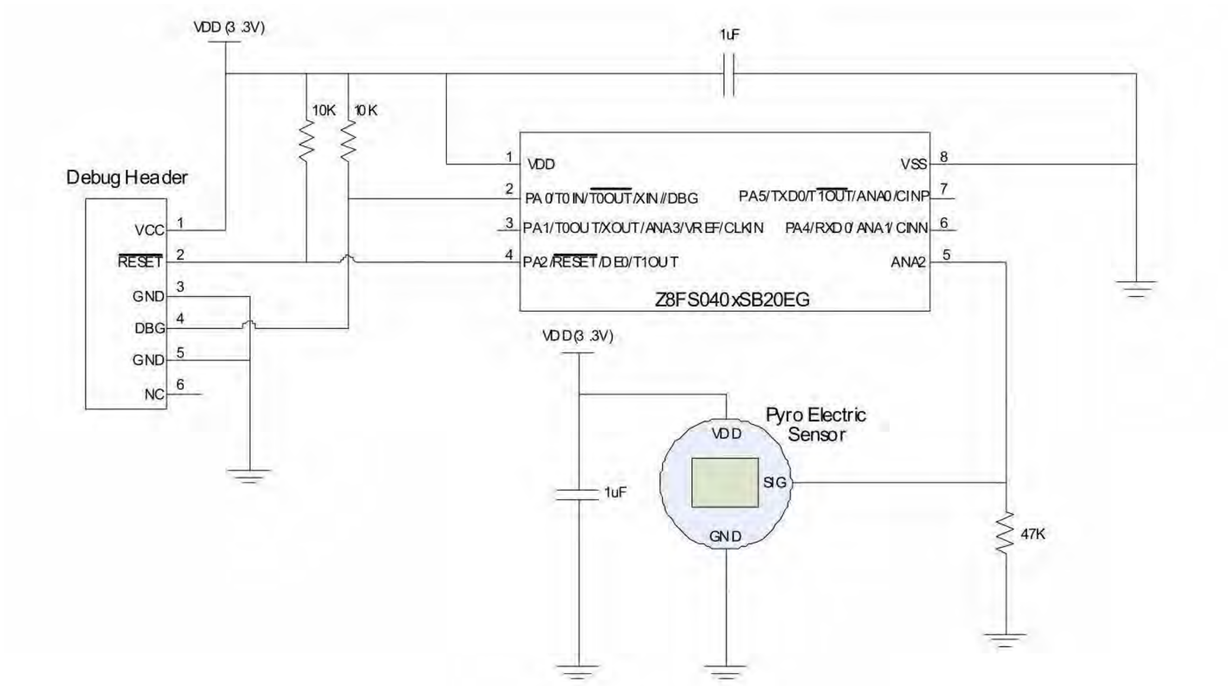

Figure 3. Required Circuit Connections for the ZMOTION Z8FS040 (8-pin) MCU

IXYS LED Driver Circuit

Figure 4 shows the LED driver circuit using the LDS8710 driver.

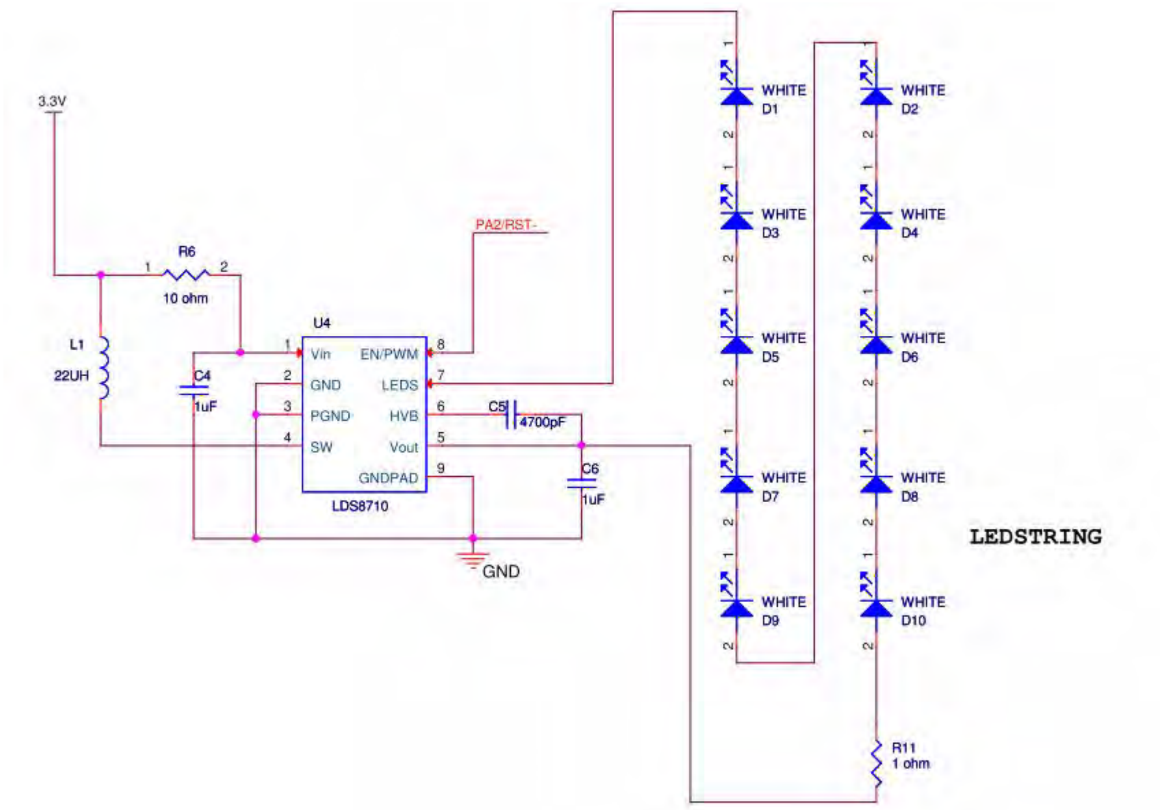

Figure 4. LDS8710 LED Driver Application Circuit

Caution: High-frequency boost converters such as the LDS8710 LED Driver generate high-speed transience on the power and ground signals. Good analog design practices must be observed to ensure that excessive noise does not affect the low-voltage signals generated by analog components such as the pyroelectric sensor or the ZMOTION MCU.

Software Implementation

The software for this reference design is based on the `ZMOTION_Serial` sample application project that ships with the ZMOTION Detection & Control Development Kit (ZMOTIONL100ZCOG). This document references three separate projects created to demonstrate the three types of LED brightness controls presented in this reference design; namely, a console command, a potentiometer setting and an ambient light sensor.

The project also includes the following three applications precompiled into hex files:

```
zmotion_ixysdisplay_console.hex
zmotion_ixysdisplay_potentiometer.hex
```

zmotion_ixysdisplay_ambientlight.hex

These three applications provide a simple demo that performs the following tasks:

- Illuminates LEDs in response to motion
- Extinguishes LEDs 10 seconds after the most recently-detected motion
- Controls light intensity in a variable manner via potentiometer, console or ambient light sensor

See Flow Charts on page 19 to review the flow of the application.

Manual LED Brightness Control via Console Commands

Use the project file and source code provided in the RD0003-SC01.zip file for this version of the application. This application allows the brightness of the LED string to be directly controlled over a serial interface; use a simple terminal program such as HyperTerminal to provide the interface. The original `ZMOTION_Serial` application code contains a set of serial commands that performs certain tasks. A *Change PWM Duty Cycle* command was added to the existing set of `ZMOTION_Serial` commands to support manual LED brightness control. The duty cycle is clipped at a 15% minimum value and a 95% maximum value, and is defined in the `rd0003.h` header file as:

```
#define LED_MAX_DUTYCYCLE:Maximum allowed duty cycle
#define LED_MIN_DUTYCYCLE:Minimum allowed duty cycle
```

Table 1 provides a complete list of the serial commands that can be performed in this reference design.

Table 1. Serial Commands

Command	Description
1) Menu	Displays the main menu.
2) Read RAM	Returns the value stores at the specified RAM address.
3) Write RAM	Changes the value stored at the specified RAM address.
4) Display Events On/Off	Toggles the motion events display.
5) Reset Event Log/Counts	Resets event log and counters.
6) Dump Event Log	Displays event logs currently in the buffer.
7) Display API Reg	Displays API register values.
8) Change PWM Duty Cycle	Changes PWM duty cycle with the user-specified duty cycle. Accepted values are in the range 00Fh–05Fh.

Table 2 lists the functions of the Z8FS040 MCU's pins when running in CONSOLE Mode.

Table 2. Z8FS040 Port Functions When Running in CONSOLE Mode

Pin Number	Port Number	Function Used	Application Function
2	PA0	Not used	
3	PA1	Not used	
4	PA2	T1OUT	PWM output for LED brightness
5	PA3	ANA2	PIR dedicated input
6	PA4	RxD	UART Rx
7	PA5	TxD	UART Tx

Manual LED Brightness Control via Potentiometer

Use the project file and source code provided in the RD0003-SC02.zip file for this version of the application. In POTENTIOMETER Mode, the serial commands and UART functions are disabled. The PWM duty cycle will change depending on potentiometer readings via ANA1. The potentiometer creates a simple voltage divider in conjunction with R3 and R5. This voltage is presented to the ANA1 input on pin 6 of the MCU. The software is configured such that as voltage increases, the PWM duty cycle increases exponentially. For details, please refer to the LED Brightness Compensation section on page 11.

Table 3 lists the Z8FS040 MCU pin functions when running in POTENTIOMETER Mode.

Table 3. Z8FS040 Port Functions When Running in POTENTIOMETER Mode

Pin Number	Port Number	Function Used	Application Function
2	PA0	Not used	
3	PA1	Not used	
4	PA2	T1OUT	PWM output for LED brightness
5	PA3	ANA2	PIR dedicated input.
6	PA4	ANA1	Potentiometer analog input
7	PA5	Not used	

Automatic LED Brightness Control via Ambient Light Sensor

Use the project file and source code provided in the RD0003-SC03.zip file for this version of the application. In AMBIENT LIGHT SENSOR Mode, the PWM duty cycle will change depending on the level of ambient light detected by the ambient light sensor via ANA3. As the amount of voltage applied to the ambient light sensor increases, the PWM duty cycle increases/decreases exponentially depending on J1 settings, as indicated in Table 4. For details, please refer to the LED Brightness Compensation section on page 11.

Table 4. J1 Settings in AMBIENT LIGHT SENSOR Mode

J1 Shunt	Operation
1–2	As ambient light voltage increases, the PWM duty cycle decreases exponentially.
2–3	As ambient light voltage increases, the PWM duty cycle increases exponentially.

Table 5 lists the Z8FS040 MCU pin functions when running in AMBIENT LIGHT SENSOR Mode.

Table 5. Z8FS040 Port Functions When Running in AMBIENT LIGHT SENSOR Mode

Pin Number	Port Number	Function Used	Application Function
2	PA0	Input	ALS increase/decrease
3	PA1	ANA3	Ambient light sensor input
4	PA2	T1OUT	PWM output for LED brightness
5	PA3	ANA2	PIR dedicated input
6	PA4	Not used	
7	PA5	Output	Ambient light sensor supply

Motion Sensitivity and Noise Reduction

Systems that include high-frequency switching elements can generate excessive noise that can disrupt low-voltage analog sensing functions. In this reference design, the change in current consumption generated by the LDS8710 driver when the LEDs are turned on or off can potentially impact the low-signal-level motion-sensing functions performed by the ZMOTION MCU. Fluctuations on the power supply or the signal from the pyroelectric sensor can cause the ZMOTION MCU to mistakenly interpret it as motion (i.e., a false trigger).

To help avoid these false triggers, there are several configuration registers included in the Z8FS040 API that can be dynamically modified as the application requires. These registers control the sensitivity and how the ZMOTION PIR Engine samples the pyroelectric sensor signal. Additionally, the application code can be written to simply ignore motion detection for short periods of time when transient or noisy conditions are known to exist. For example, when the LEDs are turned on or off (in response to motion being detected or not detected for 30 seconds), a significant disturbance in the supply voltage is generated. To ensure that false triggers are not created, the application simply stops looking for motion for 1 second after enabling/disabling the LEDs. This decision has little impact on the user’s experience, but helps avoid false triggers.

The remainder of this section presents a list of ZMOTION PIR Engine API registers that can be used to dynamically control the sensitivity and sampling of the motion detection system, along with their impact on motion sensitivity and noise reduction.

ePIR_Sensitivity. Adjusts the sensitivity of the ZMOTION PIR Engine to target motion. When set to a lower value, the Engine becomes more sensitive to motion. A typical value will range from 12 to 16.

ePIR_ASC0, Buffer Refresh Bit. Resets motion detection history to effectively ignore any possible false detection caused by fluctuations introduced to the system as external components such as LEDs turning on or off. A *buffer refresh* must be performed by setting this bit at the proper time after a fluctuation on the power supply signal is expected to occur, including the length of time that the fluctuation or noise is likely to occur. As a result, the data generated by the PIR Engine will be ignored while noise is present in the system.

ePIR_Sample_Size. Controls the amount of averaging that the Engine performs on incoming ADC samples. Higher values improve noise immunity, but at the cost of a slower sample rate.

ePIR_Debounce. Controls the amount of time that the Engine will wait to fully debounce a motion signal. Lower values require the duration of a detected motion event to be valid; higher values help to eliminate false triggers caused by short duration transience.

ePIR_SC2, Range Control. Determines the relative range of motion detected. Higher values decrease the range of detection, likewise decreasing sensitivity to lower signal changes. In a normally noisy environment in which small signal changes occur frequently – such as the 0.7MHz oscillation frequency introduced by the LDS8710 driver in this reference design – lowering the range of detection will cause the ZMOTION MCU to disregard such noise.

Power-On Stabilization Time. The ePIR_SC0 (PIR Stable Bit) is set by the Engine when it detects that the PIR sensor has stabilized. It monitors the DC offset of the PIR sensor, and sets this bit when it has become stable. The length of stabilization time depends on the PIR sensor in use and the environmental conditions under which the ZMOTION MCU is running; this stabilization time can vary from a few seconds to up to a minute. Under electrically harsh conditions, such as the high-frequency switching involved in this reference design, the PIR Engine requires more time to settle down. In other words, it requires more time to define the *normal* DC offset for this type of environment. Increasing the stabilization time after this bit is set by the Engine (i.e., adding a few more seconds) will help improve initial stability.

We can also take advantage of the flexibility provided by the ZMOTION Engine API in occupancy detection-type lighting applications by increasing the sensitivity to motion after motion has first been sensed, then returning the sensitivity to its original level (after some predetermined time) without motion being detected. As a result, the ZMOTION MCU is allowed to respond to minimal motion events while the lights are on and maintain excellent stability while the lights are off. The provided application code implements this functionality with a feature called Hypersense. There are four defines in `main.h` used to adjust the Hypersense function:

HYPER_TIME	Sets the amount of time that Hypersense stays on.
HYPER_NORMAL	Normal sensitivity is increased via the ePIR_Sensitivity Register.
HYPER_EXTENDED	Extended sensitivity is increased via the Extended Detection bits in the ePIR_SC0 Register.
HYPER_RANGE	Range is increased via the ePIR_SC2 Register's range control bits.

LED Brightness Compensation

When using a direct (linear) relationship between the input (that is, potentiometer or ambient light sensor voltage) and the output (that is, the PWM duty cycle for LED brightness control), the perceived LED light level changes in a non-linear fashion. Similarly, in this case, as the potentiometer voltage increases, the LED brightness seems to jump in steps. This is specifically visible when the LEDs are in low light level. The reason for this is that the human eye is more sensitive to changes at low light levels than it is to changes at high light levels. To compensate for this, the incremental changes to the PWM duty cycle must be more granular at lower PWM duty cycle values. That is, the PWM duty cycle must increase exponentially as the input increases.

In this reference design, a look-up table is created to achieve this exponential increase/decrease in the PWM duty cycle. The look-up table is chosen because it is much easier to calculate, plus it provides more flexibility.

Figure 5 shows the relationship between the ADC reading (from potentiometer or ambient light sensor) and the PWM duty cycle. The ADC readings were normalized to the reload value of the timer PWM to achieve a correct PWM match value for the desired PWM duty cycle.

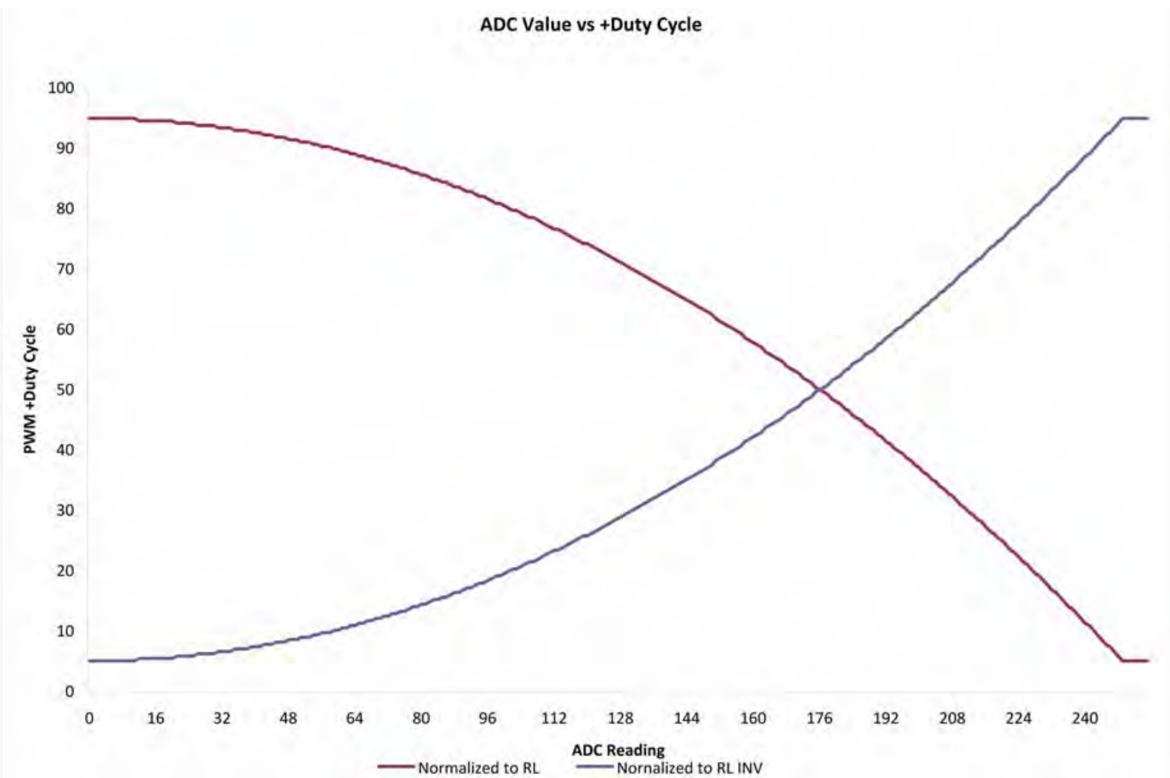

Figure 5. ADC Value (from Potentiometer/Ambient Light Sensor) vs. +Duty Cycle

Note: The look-up table associated with this reference design (*RD0003-LookupTable.xls*, contained in the RD0003-SC02.zip or RD0003-SC01.zip files), is used to calculate approximate values for the PWM match value.

Testing/Demonstrating the Application

This reference design has three modes of operation; each mode is associated with a different hex file and different jumper setting. Therefore, it is important to make sure that the correct hex file is downloaded onto the ZMOTION/IXYS Display Reference Design Board, and that the jumpers are set up properly to meet application requirements.

Table 6 lists the precompiled hex files and their associated application modes. When recompiling the source code is preferred, make sure to use the appropriate project file.

Table 6. Application Mode and Associated Hex File

Mode	Hex File
CONSOLE Mode	zmotion_ixysdisplay_console.hex
POTENTIOMETER Mode	zmotion_ixysdisplay_potentiometer.hex
AMBIENT LIGHT SENSOR Mode	zmotion_ixysdisplay_ambientlight.hex

Table 7 summarizes the jumper settings required to run this reference design properly.

Table 7. Jumper Settings for Different Application Modes

Mode	J1 Shunt	J6 Shunt
DOWNLOAD/DEBUG Mode	1–2	1–2
CONSOLE Mode	2–3	2–3
POTENTIOMETER Mode	2–3	2–3
AMBIENT LIGHT SENSOR Mode	1–2/2–3	2–3

At power-up, the LEDs are turned on at full brightness and will turn off after PIR initialization. The LEDs will turn on at a certain brightness level for 10 seconds each time motion is detected. The brightness level depends on (1) the brightness specified via the console, (2) the brightness settings identified via the potentiometer, or (3) the brightness settings determined by ambient light conditions.

Setup

The ZMOTION/IXYS Display Reference Design Board has a 2.7V–3.6V operating range that can be powered by 2 AA batteries or by an external power supply; it is up to the user's preference as to which power source to use. Take note of the switch settings to select the proper power source.

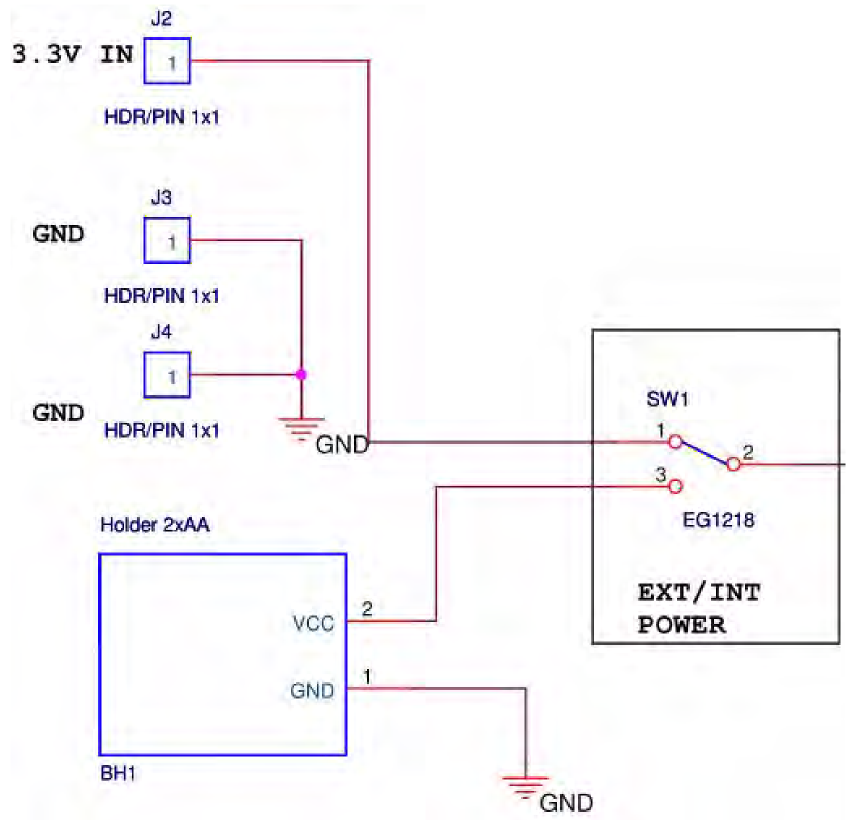

Figure 6. Switch Configuration

Caution: When downloading the application to the ZMOTION/IXYS Display Reference Design Board, a message may appear indicating that the silicon is not the latest version. This occurrence is normal and is not an issue. Click OK to continue.

Console Mode

In CONSOLE Mode, the UART interface (J8) is connected to the PC using a UART/RS-232 TTL-level converter and an RS-232 cable. HyperTerminal settings should be:

- 115200 baud rate
- 8 bits
- No parity
- 1 stop bit

- No hardware flow control

LED brightness is controlled by the *Change PWM Duty Cycle* command. See Table 1 on page 7 for a list of serial commands.

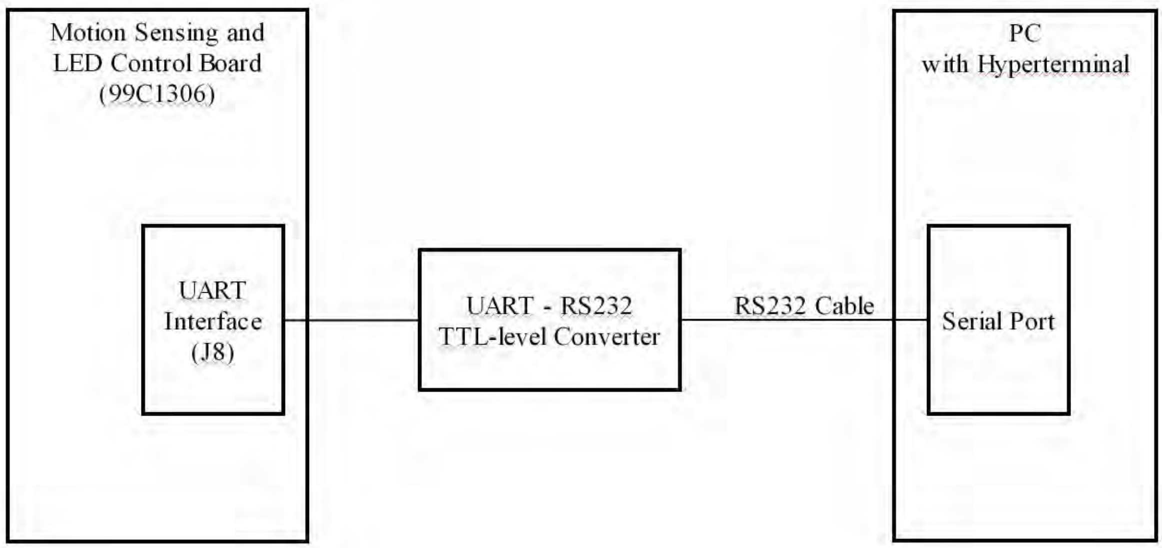

Figure 7. Hardware Setup for CONSOLE Mode

POTENTIOMETER Mode

There is no special setup requirement for POTENTIOMETER Mode. Simply power the board on to run the application. LED brightness is controlled by the potentiometer.

AMBIENT LIGHT SENSOR Mode

Operating in AMBIENT LIGHT SENSOR Mode features 2 settings, depending on J1 configuration. See Table 4 on page 9 for more J1 configuration details. LED brightness is controlled by the presence of visible light. A flashlight or a light bulb can be used for testing the ambient light sensor (see Figure 8). Holding a flashlight nearer to U1 will increase the sensor reading, while holding a flashlight farther from U1 will decrease this reading.

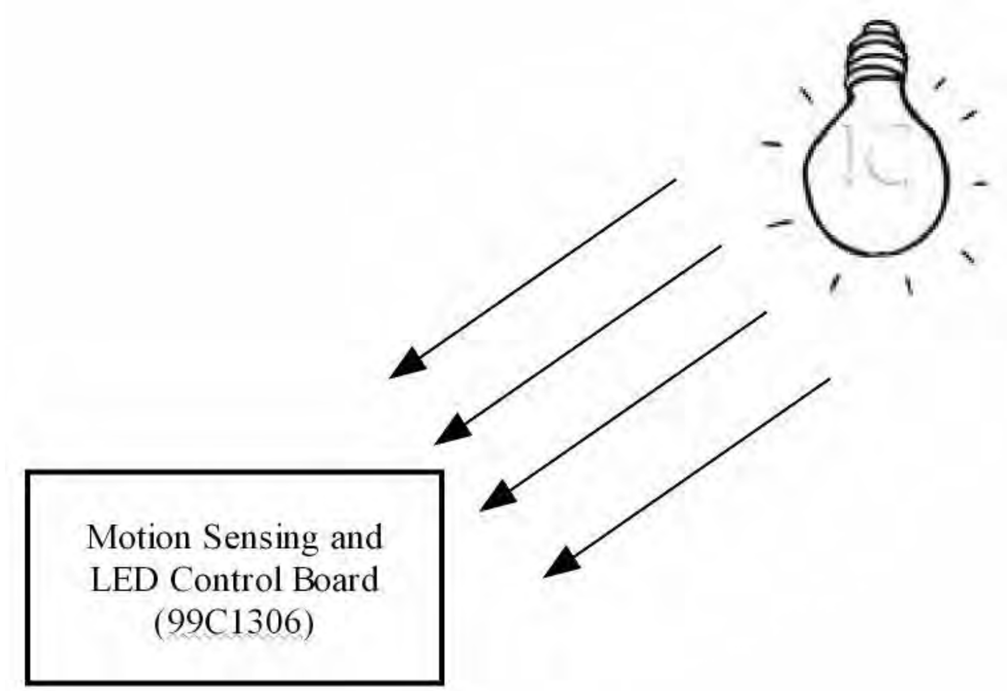

Figure 8. Hardware Setup for AMBIENT LIGHT SENSOR Mode

> **Note:** The ambient light sensor circuit ships in two versions: (1) using TEPT4400 for visible light detection and (2) using TEFT4300 for outside light detection. Please refer to Schematics on page 17 and Schematics on page 18.

Ordering Information

This Motion Sensing and LED Control Reference Design can be purchased from the Zilog Store – simply click the Store Product ID listed in Table 8.

Table 8. ZMOTION Intrusion Motion Detector Reference Design Ordering Information

Part Number	Description	Store Product ID
ZMOTIONS200ZRDG	ZMOTION Intrusion Motion Detector Reference Design	RD10012

Related Documentation

The documents associated with this Motion Sensing and LED Control Reference Design are listed in Table 9. Each of these documents can be obtained from the Zilog or IXYS websites by clicking the link associated with its document number.

Table 9. ZMOTION Intrusion Motion Detector Reference Design Documentation

Document Number	Description
UM0128	eZ8 CPU User Manual
PS0228	Z8 Encore! XP F082A Series Product Specification
PS0286	ZMOTION Lens and Pyroelectric Sensor Product Specification
WP0018	ZMOTION Detection Lens and Pyro Sensor Configuration Guide
PS0285	ZMOTION Detection and Control Product Specification
LDS8710	High-Efficiency 10 LED Driver With No External Schottky

Appendix A. Schematics

Figure 9 shows a schematic diagram representing the ZMOTION/IXYS Display reference design using a TEPT4400 ambient light sensor circuit for visible light detection.

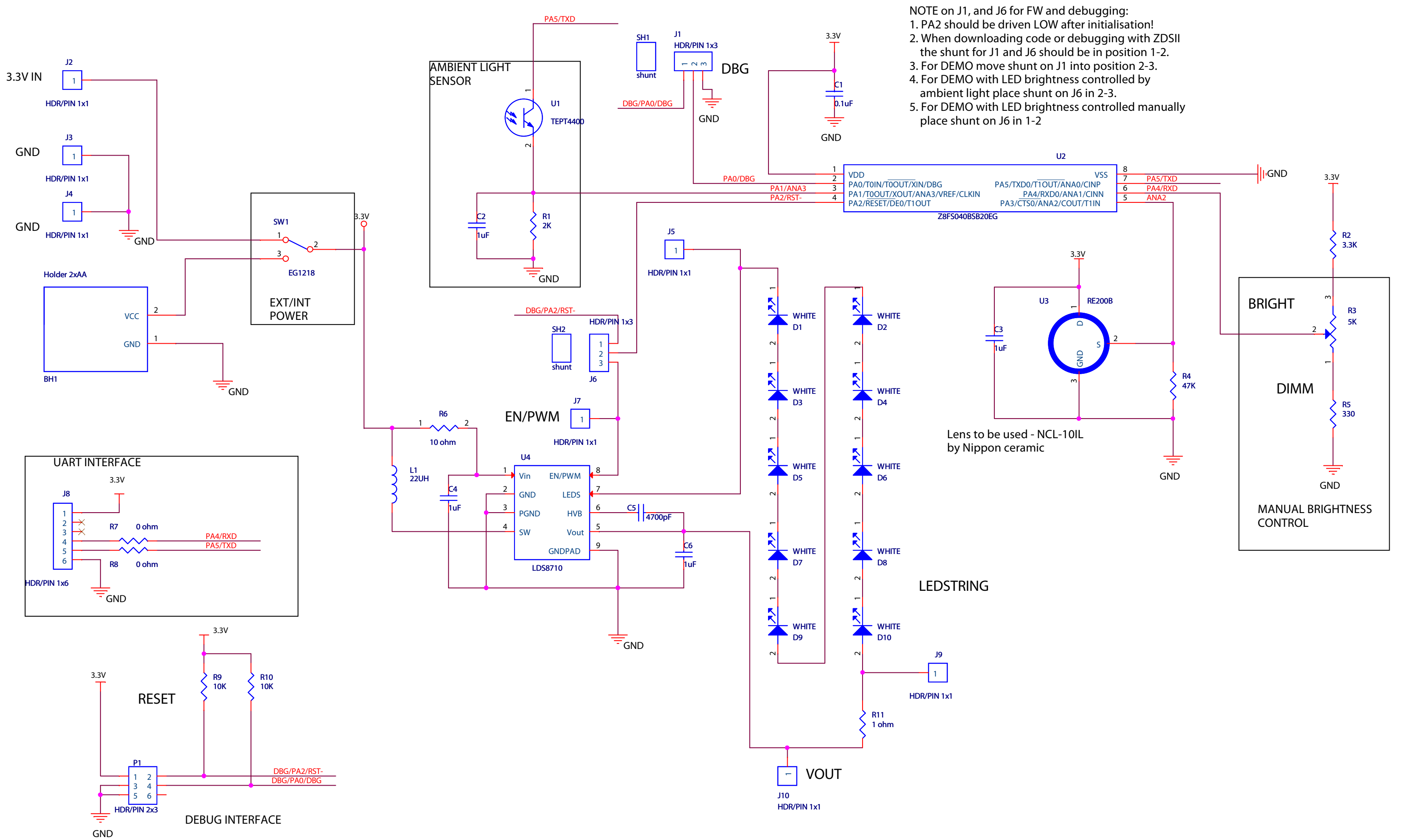

Figure 9. ZMOTION/IXYS Display (Visible Light) Schematic Diagram

Appendix B. Schematics

Figure 10 shows a schematic diagram representing the ZMOTION/IXYS Display reference design using a TEPT4300 ambient light sensor circuit for natural (outside) light detection.

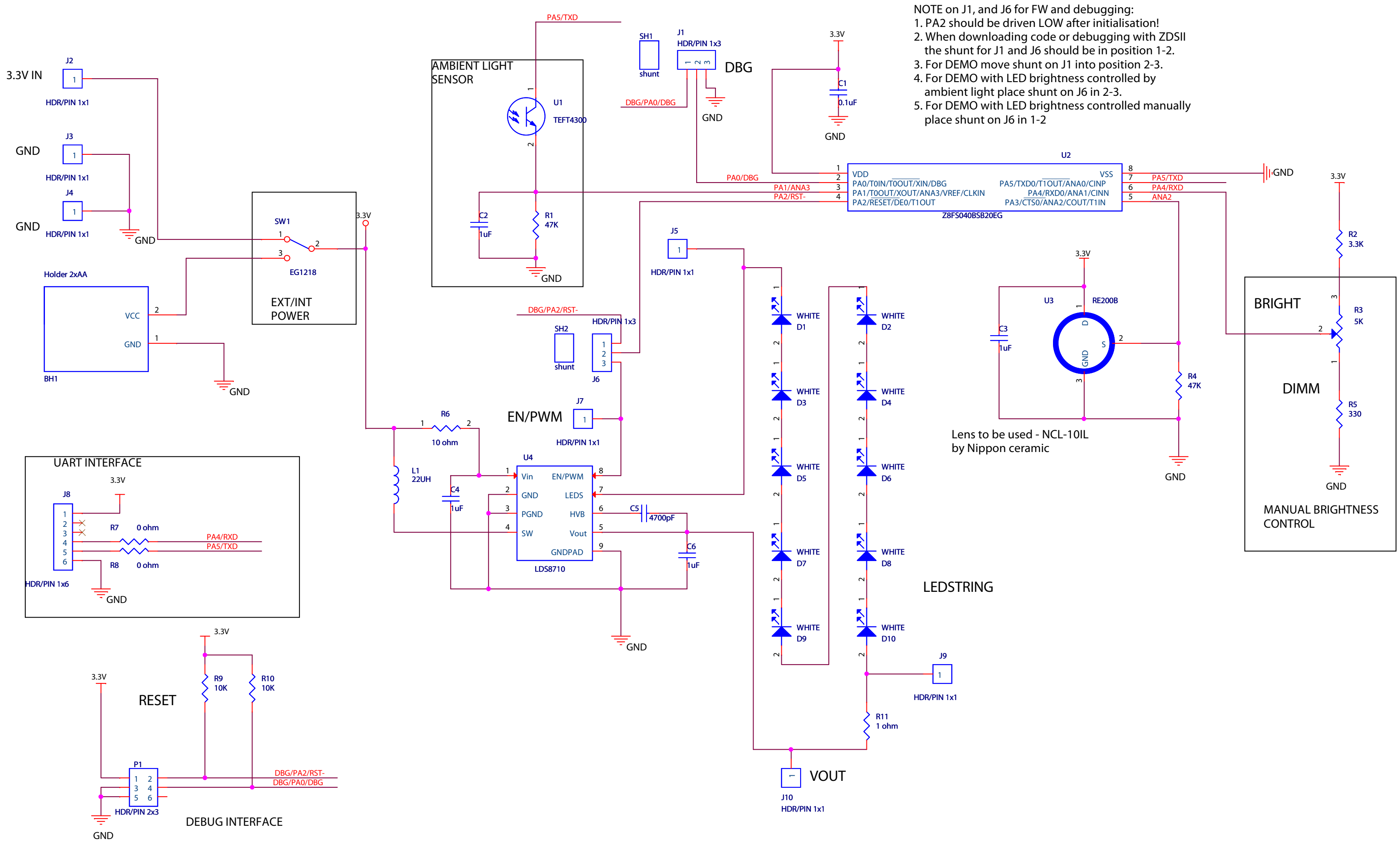

Figure 10. ZMOTION/IXYS Display (Outside Light) Schematic Diagram

Appendix C. Flow Charts

Figure 11 displays the sequence of the `main.c` application code.

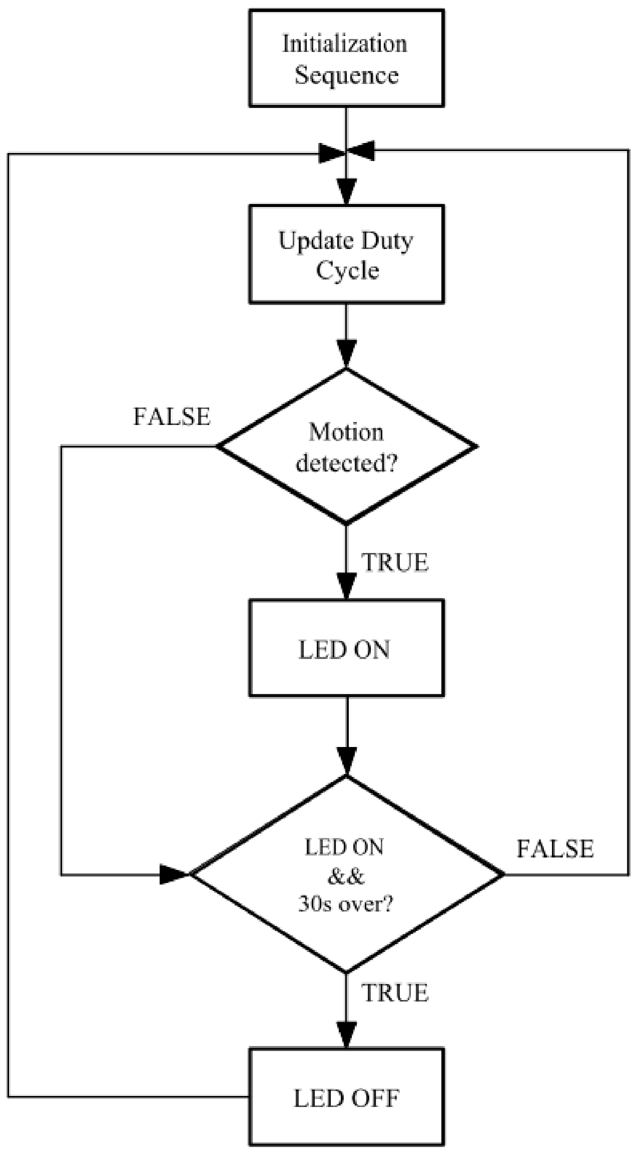

Figure 11. Main Application Flow

Figure 12 displays the `main.c` initialization sequence.

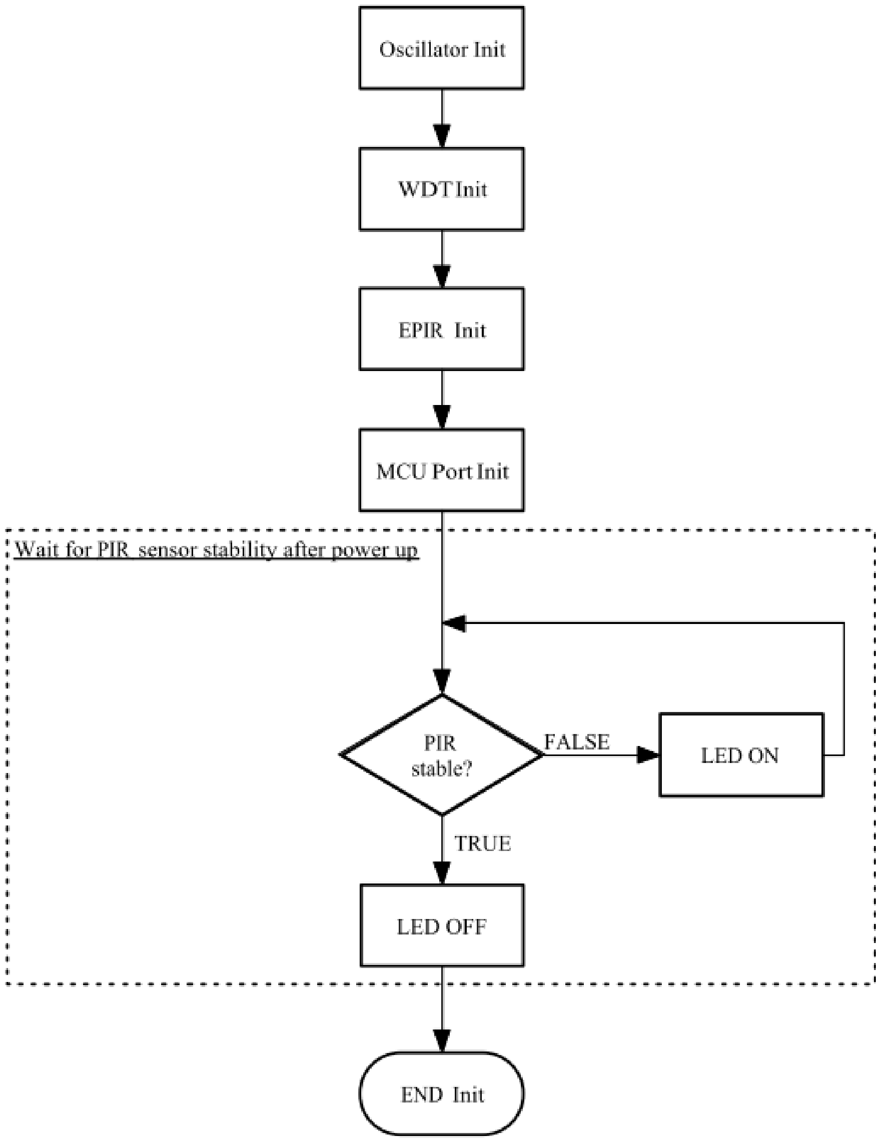

Figure 12. Initialization Sequence

Figure 13 displays the flow of the PWM duty cycle routine.

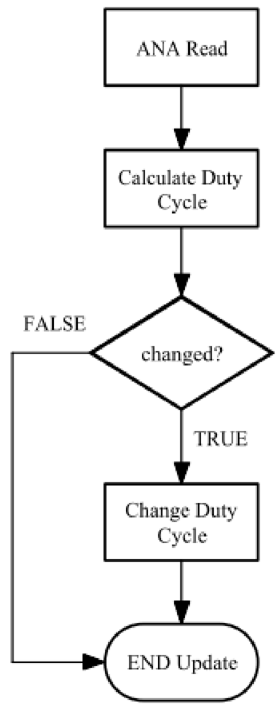

Figure 13. Update PWM Duty Cycle Flow

Customer Support

To learn more about this product, find additional documentation, get your technical questions answered or report issues, please contact esales@zilog.com.

Warning: DO NOT USE THIS PRODUCT IN LIFE SUPPORT SYSTEMS.

LIFE SUPPORT POLICY

ZILOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer