Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

THIS SPEC IS OBSOLETE

Spec No: 002-05438

Spec Title: AN205438 - F2MC-8FX FAMILY MB95260 SERIES 8-BIT MICROCONTROLLER ULTRASONIC DISTANCE MEASUREMENT

Replaced by: NONE

AN205438

F²MC-8FX Family MB95260 Series 8-Bit Microcontroller Ultrasonic Distance Measurement

Associated Part Family: MB95260 Series

This application note describes how to implement distance measurement based on ultrasonic sensors.

OBSOLETE

Contents

1 Introduction....1
1.1 Features....1
1.2 Functional Principle....1
2 Hardware....2
2.1 Ultrasonic and Analog Circuit....3
2.2 Microcontroller....4
3 Software....5
3.1 Main Code....6
3.2 USART Interface....7
3.3 Flash Writing....8
3.4 Timing Routines....8
4 Interface Protocol....9
4.1 Packages Sent by the Ultrasonic Module....9
4.2 Commands....10
5 Summary....13
6 Additional Information....13
Document History....14

1 Introduction

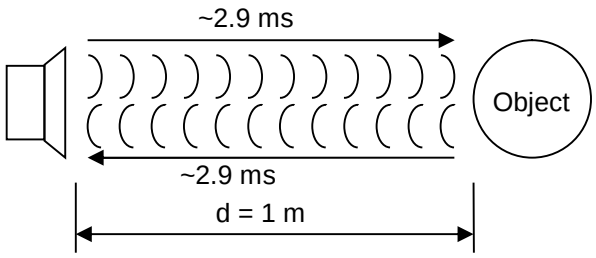
This application note describes how to implement distance measurement based on ultrasonic sensors. It will give both an example hardware and software implementation with a Cypress 8-bit microcontroller.

1.1 Features

- Measure distances from about 20 to 400 cm
- Storage of calibration data in internal Flash
- USART as communication interface
- Standalone operation possible
- Single 5 V power supply

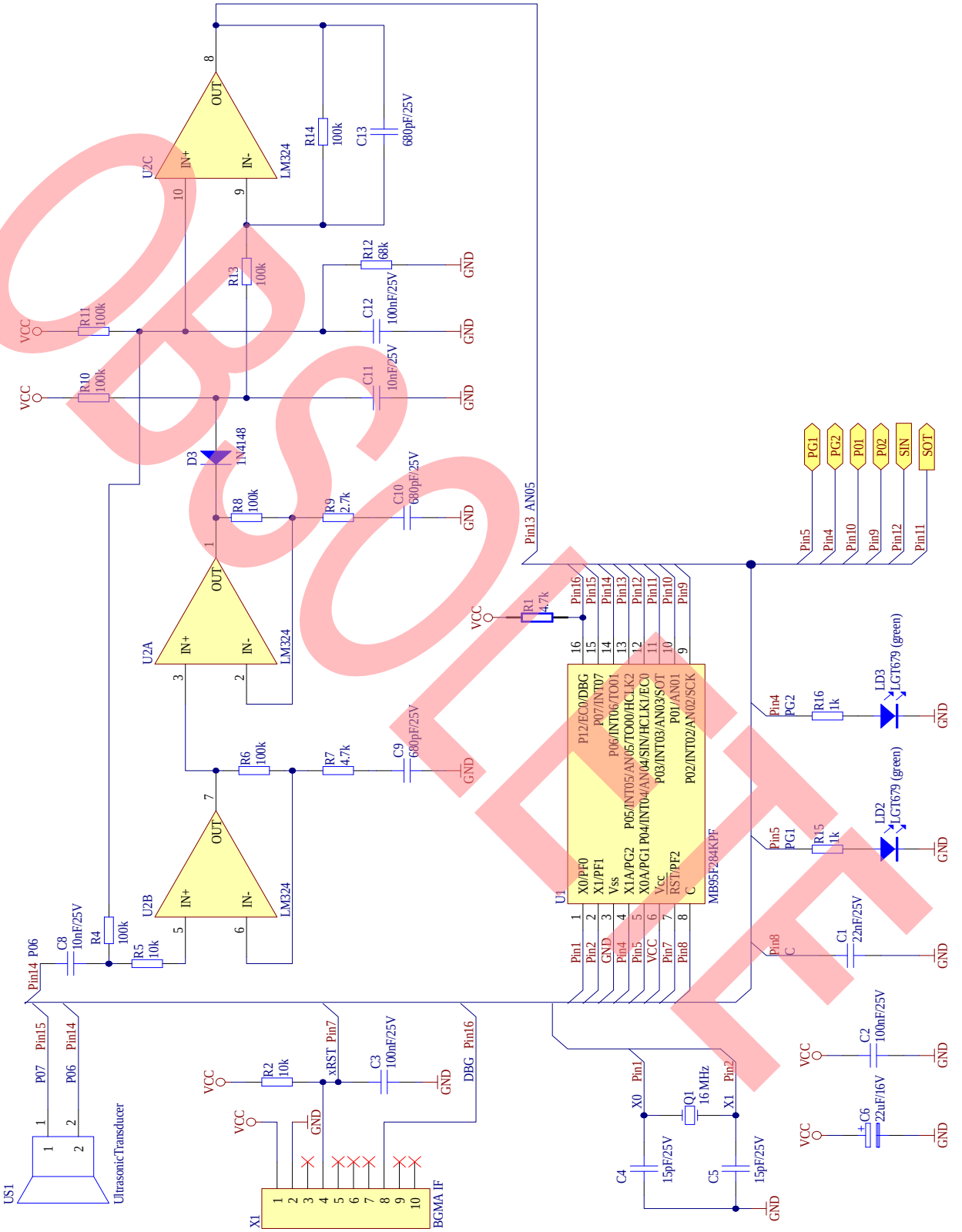
1.2 Functional Principle

The distance measurement with ultrasonic modules is based on the reflection of sound waves. An ultrasonic wave, i.e. a sound with a high frequency of normally about 40 kHz, is generated in a speaker which is also called ultrasonic transducer. This wave is reflected by objects. When the reflected wave reaches the ultrasonic transducer it converts the energy back into electrical voltage which can be measured. The time between sending the signal and receiving the reflected signal can then be used to calculate the distance of the reflecting object based on the velocity of sound.



Total signal propagation time (t): ~ 5.8 ms
Signal propagation time one way: ~ 2.9 ms
Velocity of sound (v): ~340 m/s
Distance (d):
d = v * t = 340 m/s * 2.9 ms = 0.99 m

2 Hardware



OBSOLETE

The schematic consists of two functional blocks:

- Ultrasonic and analog circuit
- Microcontroller circuit

2.1 Ultrasonic and Analog Circuit

The pins of the ultrasonic transducer US1 are attached to two IO pins of the microcontroller. Sending the ultrasonic bursts is done by setting one pin to high while the other is low and vice versa. This way a +/- 5 V square wave signal is generated. When sending is done one pin of the ultrasonic transducer, connected to P07 (pin 15) of the microcontroller, is set to ground so the received power from the ultrasonic signal can be measured as a voltage on the second pin. The connected microcontroller IO P06 (pin 14) is set to high impedance so the signal can be amplified through the operational amplifier circuit. It consists of four blocks:

- Coupling capacitor
- Non-inverting amplifier 1
- Non-inverting amplifier 2
- Active low-pass filter

The amplified signal is measured by the analog input AN05 (pin 13) of the microcontroller.

2.1.1 Coupling Capacitor

The coupling capacitor (C8) is used to couple the AC signal from the ultrasonic transducer on a 2 V DC voltage generated by the voltage divider R11/R12. This is needed to be able to use the single-supply operational amplifiers.

2.1.2 Non-inverting Amplifiers

The first amplifier stage consists of U2B, R6, R7 and C9, the second one of U2A, R8, R9 and C10.

Their amplification can be calculated as:

$$v_1 = 1 + R6 / R7 = 1 + (100\ k\Omega) / (4.7\ k\Omega) \approx 22.3$$

$$v_2 = 1 + R8 / R9 = 1 + (100\ k\Omega) / (2.7\ k\Omega) \approx 38.0$$

C9 and C10 are required to reduce DC amplification of the operational amplifiers.

2.1.3 Active Low-pass Filter

The low-pass filter is used to smooth the half-wave signal coming from D3 after being amplified by U2A. Its amplification is:

$$v_3 = - (R14 / R13) * 1 / (1 + j * \omega * C13 * R14) =$$
$$- (R14 / R13) * 1 / (1 + j * 2 * \pi * f * C13 * R14) =$$
$$- (100\ k\Omega / 100\ k\Omega) * 1 / (1 + j * 2\ \pi * 40\ kHz * 680\ pF * 100\ k\Omega) \approx 0.06$$

The cutoff frequency of the filter is:

$$f_c = 1 / (2 * \pi * R14 * C13) = 1 / (2 * \pi * 100\ k\Omega * 680\ pF) \approx 2341\ Hz$$

OBSOLETE

2.1.4 Total Gain

Total gain of the amplification circuit is calculated by:

$$v_t = v_1 * v_2 * v_3 = 22.3 * 38.0 * 0.06 = 49.6$$

2.2 Microcontroller

The microcontroller (U1) is a Cypress's MB95F284K 8-bit microcontroller. It is used to generate the 40 kHz ultrasonic bursts and to measure the time until the echo is received.

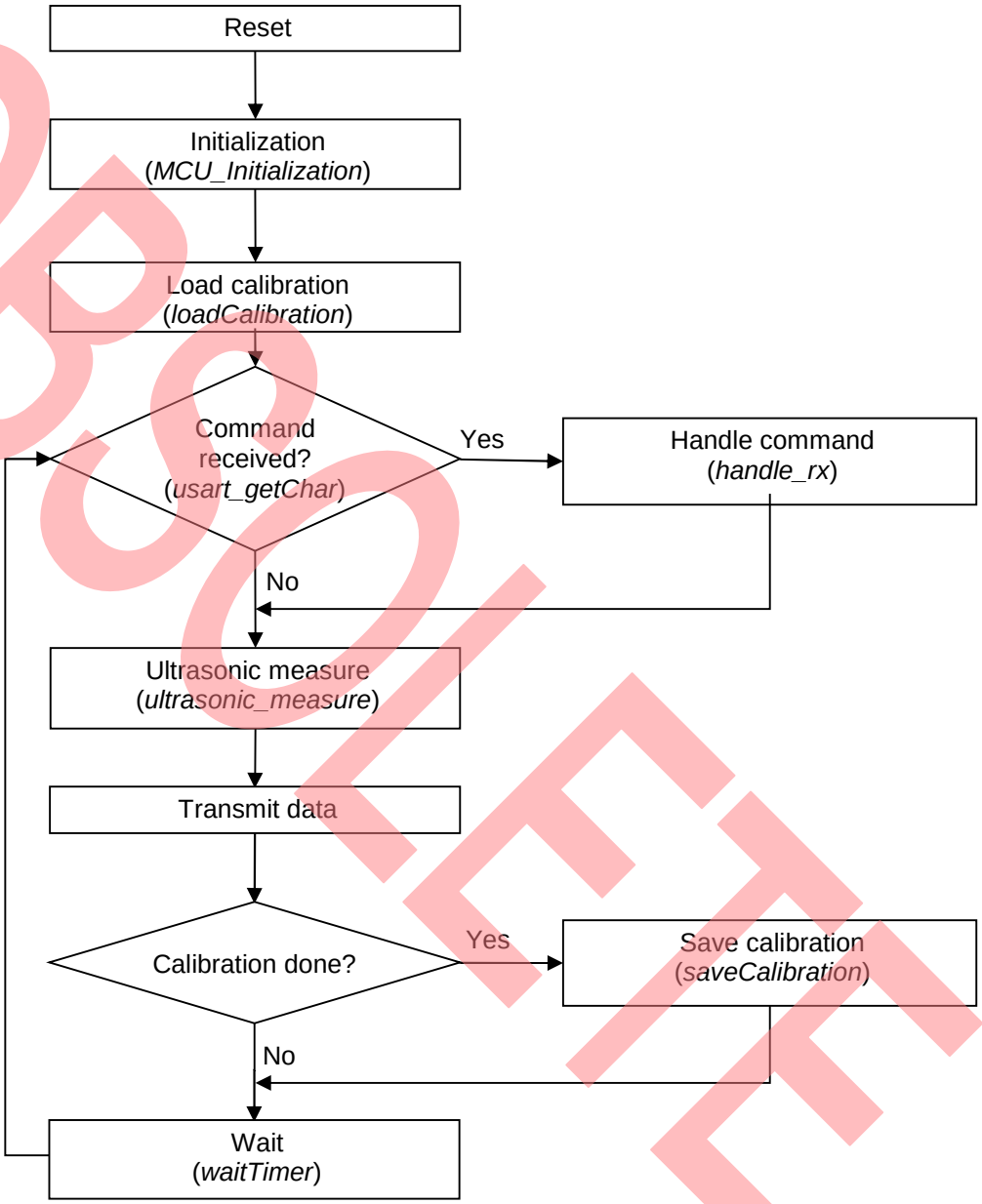
The clock is provided with a 16 MHz crystal oscillator resulting in 8 MHz CPU clock which allows a measurement resolution of 128 µs or about 2 cm. LEDs are attached to pins PG1 and PG2 which can be either controlled through the USART interface or for standalone operation by enhancing the software. Pins P01, P02, PG1 and PG2 can be used as GPIOs. SIN and SOT provide the control interface. Connector X1 provides the debug interface for the BGM-Adapter MB2146-08-E.

OBSOLETE

3 Software

The firmware on the ultrasonic module (95280_adc_ultrasonic) samples the echo of the ultrasonic transmission and provides an evaluated output of that data which gives the distance to a detected object.

Figure 1. Flowchart of Ultrasonic Module Firmware



OBSOLETE

The measurement routine first sends 13 ultrasonic bursts. Then it converts 8 samples of the returned signal which are averaged in each of 250 timeslots. From the averaged value, the stored calibration value is subtracted. The timeslots are 128 µs each which results in about 2 cm of additional distance per timeslot until a peak is detected.

The code is split into four modules:

- Main code including protocol handler and ultrasonic measurement (main.c)
- USART interface (usart.c)
- Flash writing (flash.c)
- Timing routine (util.c)

3.1 Main Code

Main code first initializes the used hardware macros (routine MCU_Initialization):

- Selecting the external crystal as main clock source
- Set Timebase Timer to 128 µs interval
- Analog/digital converter (AN05)
- USART interface
- Pins for the ultrasonic transducer output (P06, P07)
- User I/O pins (P01, P02, PG1, PG2)

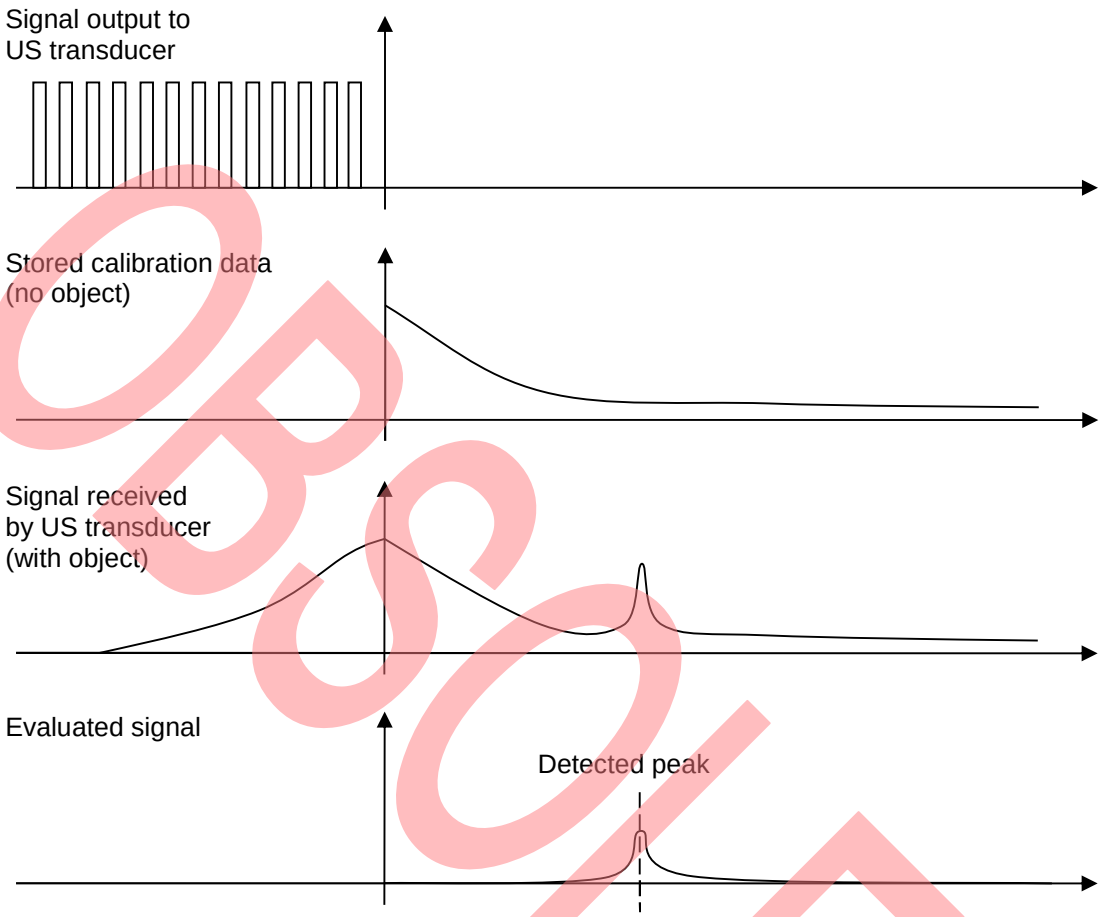
Next step is to load the calibration data from the lower Flash bank (loadCalibration).

At this point the CPU enters the infinite loop which does the following:

- Check whether a command was received via USART (usart_getChar).
  - If so try to handle the received byte (handle_rx).
- Do one measurement run (ultrasonic_measure).
  - Send the ultrasonic burst (sendUSburst).
  - Get 250 samples from the analog converter in 128 µs interval.
  - Return the index of the first detected peak.
- Save calibration data if a calibration was just finished (saveCalibration).
- Wait for a configurable multiple of 50 ms (see 4.2.2. Pause).

OBSOLETE

Figure 2. Ultrasonic Measurement Process



OBSOLETE

3.2 USART Interface

This module (usart.c) provides methods to initialize the USART, send bytes, decimal values and strings and receive bytes.

- usart_init : initializes the USART to asynchronous 8N1 with 1 MBaud.
- usart_putChar : sends a single byte.
- usart_putString : sends a null-terminated string.
- usart_putDec : sends a decimal value between 000 and 999.
- usart_putDecInt: sends a decimal value between 00000 and 65535.
- usart_getChar : gets a single byte if USART receive register is not empty.

Since the interface is running at 1 MBaud it is not usable with a normal RS232 interface of a PC. It can be used through USB to serial adapters though. The baud rate is set to 1 MBaud in order to reduce the time needed to transfer the data while measuring.

3.3 Flash Writing

This module (flash.c) is used to erase the first sector of lower flash bank (SA0, starting at $B000_H$) and write bytes to it.

- flash_erase : erases the sector containing the specified address. Used to remove stored calibration data in “clearCalibration”.
- flash_write : writes a single byte to a Flash address. Writing to flash is used by “saveCalibration” to permanently store calibration data.

Flash Bank	Sector Name	Start/End Address
Lower flash bank	SA0	$B000_H$ $B7FF_H$
	SA1	$B800_H$ $BFFF_H$
Upper flash bank	SA2	$C000_H$ $FFFF_H$

3.4 Timing Routines

This module (util.c) provides timing routines to precisely control execution of measurement operations.

- timer_init : initializes the Timebase Timer for 128 µs ticks.
- waitTimer : waits the specified amount of ticks.
- wait10times : inserts 10 NOP instructions.
- timer_delay : waits the specified times multiplied by 50 ms.

The source code of the methods “waitTimer” and “wait10times” are not included in util.c but in main.c because they are in lined in other methods to remove function call overhead.

OBSOLETE

4 Interface Protocol

The firmware features a small command protocol to control the behavior of the ultrasonic module. This protocol will be explained in this section.

4.1 Packages Sent by the Ultrasonic Module

Data sent from the module is fit into packages. Each package is prefixed with a 0 value byte whereas no other 0 value will be in the data bytes of the package. The second byte in a package determines the type of the package.

Table 1. Ultrasonic module package structure

Byte 0	Byte 1	Byte 2..n
0x00	Type of package	Data; length depending on type of package

The following types of packages are defined and will be described in the following sections:

Table 2. Ultrasonic module package types

Type value	Data length n	Description
0x01	250	Raw sample data
0x02	3	Evaluated measurement values
0x03	1	Reply to command
0x04	1	Calibration saved or unknown command

4.1.1 Raw Sample Data

This package is sent during each measurement run if the raw data mode is enabled (see 4.2.1. Raw Data Mode). It represents the averaged sample values of the analog to digital converter per timeslot. Each of the bytes is one average value.

Byte 0	Byte 1	Byte 2..251
0x00	0x01	Averaged AD sample values

The first value is for timeslot 0 which starts at about 581 us. The value for timeslot 0 is always 1.

4.1.2 Evaluated measurement values

This package is sent after each measurement run if raw mode is disabled. It contains the timeslot index and the distance in cm where an object was detected.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
0x00	0x02	Index of timeslot	Upper 7 bits of distance	Lower 7 bits of distance

The bytes for the distance in cm are OR'ed with 0x80 and the lower 7 bits of each byte contain the actual value. Thus the distance in cm can be taken from bytes 3 and 4 by:

Distance [cm] = ((Byte3 and 0x7f) << 7) or (Byte4 and 0x7f)

OBSOLETE

4.1.3 Reply to Command

If a command was received and could be handled correctly, a reply will be generated. This reply contains the original command. If the command was a read command, the value fields will be filled by the current internal values of the running software.

Byte 0	Byte 1	Byte 2
0x00	0x03	Handled command

4.1.4 Calibration Saved or Unknown Command

When a calibrate command was received and calibration is done or when an unknown command was received this package will be sent.

Byte 0	Byte 1	Byte 2
0x00	0x04	0x10 – Calibration done 0x11 – Unknown command

4.2 Commands

Commands are single bytes sent to the ultrasonic module. Since only one command is handled after each measurement run, then only one command may be sent until the reply is received. Alternatively a wait time may be used to make sure that the command has been processed.

Commands have the following format:

Table 3. Structure of ultrasonic commands

Bit 7..4	Bit 3	Bit 2..0
Parameter data	Read (0) / Write (1)	Command id

The commands defined are as following:

Table 4. Ultrasonic commands

Command id	Description	Remarks
0	Raw data mode	
1	Pause between measurement runs in units of 50 ms	
2	Calibrate to current measurement values	Write only
3	Clear calibration	Write only
4	Pin data (PDR register)	
5	Pin configuration (DDR and PUL registers)	
6	Reserved	
7	Reserved	

OBSOLETE

4.2.1 Raw Data Mode

This command allows selection of raw data mode or evaluated data mode.

Bit							
7	6	5	4	3	2	1	0
1	-	-	Mode	R/W	0	0	0

Mode bit	Mode description
0	Evaluated data mode
1	Raw data mode

4.2.2 Pause

A pause between each measurement run can be set from 0 to 750 ms.

Bit							
7	6	5	4	3	2	1	0
Number of 50 ms pause states				R/W	0	0	1

Default value is 0 ms delay.

4.2.3 Calibrate

Stores the current measurement values as calibration data in the Flash of the 8 bit controller. These values are subtracted from the measured values in measurements after this command has finished.

Bit							
7	6	5	4	3	2	1	0
1	0	1	0	1	0	1	0

When calibration has finished a “calibration saved” package (see 4.1.4. Calibration Saved or Unknown Command) is sent by the ultrasonic module.

Note: Calibration should be issued while the ultrasonic transducers are directed at an empty room so a good null measurement value can be taken.

4.2.4 Clear Calibration

This command clears stored calibration data setting every value to 0. This way the sampled data can be received in an unaltered way in raw data mode.

Bit							
7	6	5	4	3	2	1	0
0	1	1	0	1	0	1	1

Note: After this command was issued evaluated data mode will not return useful data.

OBSOLETE

4.2.5 Pin Data

This command is used to write to or read from the PDR registers of the unused IO pins of the microcontroller.

Bit							
7	6	5	4	3	2	1	0
-	Value	Pin1	Pin0	R/W	1	0	0

Pin1..0 Bits	Pin
00	PG1
01	PG2
10	P01
11	P02

4.2.6 Pin Configuration

With this command the DDR (data direction) and PUL (pull-up enable) registers can be accessed.

Bit							
7	6	5	4	3	2	1	0
Register	Value	Pin1	Pin0	R/W	1	0	1

Pin1..0 are the same values as for the pin data command.

Register Bit	Register
0	DDR
1	PUL

Value Bit	DDR Register Meaning	PUL Register Meaning
0	Input pin	Pull-up disabled
1	Output pin	Pull-up enabled

4.2.7 Reserved IDs

These two command ids are reserved for future extensions.

Bit							
7	6	5	4	3	2	1	0
-	-	-	-	-	1	1	0
-	-	-	-	-	1	1	1

OBSOLETE

5 Summary

This application note describes a functional but basic implementation for distance measurement based on ultrasonic modules. Improvements could be:

- Completely standalone operation without the need of another controller reading out distance values.
- Increased resolution of measurement.
- Increased maximum measurable distance.
- Improved echo peak detection algorithm.

6 Additional Information

For more information on Cypress products, please visit our website:

http://www.cypress.com/cypress-microcontrollers

The software example of this application note is:

95280_adc_ultrasonic

It can be found on the following web page:

http://www.cypress.com/documentation/software-and-drivers/95280-adc-ultrasonic-v10

OBSOLETE

Document History

Document Title: AN205438 – F²MC-8FX Family MB95260 Series 8-Bit Microcontroller Ultrasonic Distance Measurement

Document Number: 002-05438

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	-	WOFR	02/25/2010	Initial release
*A	5297463	WOFR	06/07/2016	Migrated Spansion Application Note MCU-AN-300119-E-V10 to Cypress format.
*B	5874545	AESATMP8	09/05/2017	Updated logo and Copyright.
*C	6019687	MBGR	01/08/2018	Making obsolete at WOFR's request.

OBSOLETE

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

OBSOLETE

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2010-2018. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.