INTRODUCTION

The Digitally Addressable Lighting Interface (DALI) has emerged as a standard in Europe to address growing power issues, mostly for commercial and industrial purposes. DALI is part of the IEC 60929 specifications, and specifically relates to digitally controlled dimmable fluorescent ballasts.

This document is written in conjunction with the DALI Communication application note (AN1465A), (www1.microchip.com/downloads/en/AppNotes/01465A.pdf), which describes the basic communication between the control device and the control gear, including explanation of electrical specifications, timing, packet formats and Manchester encoding/decoding.

This application note explains the implementation of some DALI commands on the control gear, how commissioning is performed, as well as the modes of operation, and how easy it is to add new commands to the library.

TERMINOLOGY

- Control Gear: Ballast or Sensor/Receiver
- Control Device: Controller/Transmitter
- Forward Frame: Packet sent from the control device to the control gear
- Backward Frame: Response packet sent from the control gear to the control device
- Short Address: Address of an individual control gear in the system
- Group Address: Address to a group of control gear
- Broadcast: Address used to send commands to all the control gears at once
- Te: Half the information Rate (1200 bits/sec): 416.67 µsec

DALI COMMANDS

For a complete set and detailed explanation of DALI commands, please refer to International Standard IEC 62386-102, Edition 1.0 2009-06.

DALI commands implemented in the control gear include:
- Direct arc power command
- Indirect arc power control commands (0 to 15)
- Configuration commands (32 to 128)
- Query commands (144 to 196)
- Special commands (256 to 270)

MODES OF OPERATION

Listen Mode

The control gear will stay in Listen mode for 2Te before going into the Receive mode. This does not include the idle time of 4Te, which is part of the Stop bit of an incoming packet. Once the Receive or Transmit is completed, the control gear returns to the Listen mode. In the event of an error in reception or transmission, the control gear is reset to Listen mode as well.

Receive Mode

If the command received during this mode is not meant for the control gear (the short address mismatch), the frame will be discarded and the control gear will return to Listen mode immediately. If the command is meant for the control gear (short address match), the command is extracted from the packet and proper action is taken. If the action requires the backward frame to be sent back, the next mode is Transmit mode, otherwise the next mode is Listen mode.

Transmit Mode

As the name indicates, the gear in Transmit mode will send a reply in the form of the backward frame to the control device. The backward frame is made by a Start bit, an 8-bit reply and two Stop bits (idle line). The 8-bit data is either a ‘Yes’ (0b11111111) or 8-bit information (0b11111111). Once the packet is sent successfully, the control device returns to Listen mode again.
In the Event of an Error

In the event of an error in reception, the control gear is sent back to Listen mode, and the packet is ignored. While commissioning, there is a packet collision due to multiple control gears responding with a yes (0xFF) simultaneously to a certain random address sent by the control device. In this case, the error is handled as a yes (0xFF). However, if the commissioning process is not under way, then any collision will be considered an error, and the frame will be discarded without being processed.

The following diagram (Figure 1), explains the change of modes in a control gear.

FIGURE 1: COMMUNICATION MODES

COMMISSIONING

Commissioning is a process of assigning a unique short address to each control gear. This short address is needed for sending individual commands to specific control gear, and also for sending back replies from the control gear to the control device, if needed.

The process of commissioning is started by the control device. Each control gear generates its own 24-bit random address. As the control device finds these addresses one at a time, the user chooses the short address of the ballast by some sort of push button or option menu system. This short address is then saved in the flash on the ballast. Once a short address is assigned to a control gear, the address is taken off of the list of short addresses to be used for commissioning. So, if more control gear is added and commissioned in the future, the addresses already taken will not be reused.

Commissioning is handled by the special commands (258 to 270) described in the DALI specifications (International Standard IEC 62386-102, Edition 1.0 2009-06).
Commissioning Timing

Initialize command (258) and Randomize command (259) are sent twice each within 100 ms before either can be executed, to reduce the probability of incorrect reception. No other commands addressing the control gear can be sent between the reception of two successive Initialize or Randomize commands. If a different command is received after reception of either the first Initialize or the first Randomize command, the first command received will be ignored.

The Initialize command (258) triggers a 15-minute timer. The commands 259 to 270, if received, shall only be executed while this 15-minute timer is still running. The sequence of commands received by the control gear and the responses sent back to the control device, during the commissioning process, can be understood better by the following diagram (Figure 2).

FIGURE 2: CONTROL GEAR COMMISSIONING FLOWCHART
ADDING COMMANDS TO THE DALI LIBRARY

It is very simple to add additional commands to the DALI Library. The commands have to be added to the DALI_Commands.c file. As the control gear receives any packet, the first thing it does is to identify the type of command. This information is added in the IdentifyReceiveFrame() function in the DALI_Commands.c file. Make sure the type of command you are about to add is already being checked for. If it is not there already, you will have to add it, and then call your command processing function from there. If you do not already have a command processing function the new command will be a part of, then you will have to write that function and call it from within the IdentifyReceiveFrame() function.

The following example shows that, if a command with FwdFrame.Command.Type<4 is received, then it has to be checked further to see whether it is a direct or indirect command, then the function to process the command is called accordingly, i.e., ProcessIndirectCommand() or DirectArcPowerCommand().

EXAMPLE 1: ADDING COMMANDS

```c
void IdentifyReceivedFrame(void)
{
    static uint8_t i = 0;
    FrameType.AllFrameTypes = CLEAR;

    if (FwdFrame.Command.Type < 4) //Individual Address
    {
        FrameReceived.Byte.Data = FwdFrame.Byte.Data;
        FrameReceived.Byte.Address = FwdFrame.Byte.Address;
        //Check to see if this is our address
        if(FrameReceived.Bits.Address == ShortAddress)
        {
            if (FrameReceived.Bits.Indirect)
            {
                FrameType.Indirect_ArcPower_Command = SET;
                ProcessIndirectCommand();
            }
            else
            {
                FrameType.Direct_ArcPower_Command = SET;
                DirectArcPowerCommand();
            }
        }
    }
}
```

CONCLUSION

DALI commissioning and the bulk of commands described in IEC 62386-102 have been implemented in the control gear ‘C’ code. This code provides a solid foundation to further enhance DALI features, and includes any additional commands that may be required for any development in the future.

REFERENCES

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip’s Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Microchip makes no representations or warranties of any kind, whether express or implied, written or oral, statutory or otherwise, related to the information, including but not limited to its condition, quality, performance, merchantability or fitness for purpose. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademarks
The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, KEELoQ, KEELoQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC® logo, rPIC, SST, SST logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.
FilterLab, Hampshire, Hi-TECH C, Linear Active Thermistor, MTP, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.
Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.
Analog-for-the-Digital Age, Application Maestro, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICKit, PICtail, REAL ICE, rFLAB, Select Mode, SQI, Serial Quad I/O, Total Endurance, TSHARC, UniWinDriver, WiperLock, ZENA and Z-Scale are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.
SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.
GestIC and ULPP are registered trademarks of Microchip Technology Germany II GmbH & Co. & KG, a subsidiary of Microchip Technology Inc., in other countries.
All other trademarks mentioned herein are property of their respective companies.
© 2012, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.
Printed on recycled paper.
ISBN: 9781620768099

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
== ISO/TS 16949 ==