Microchip MCP4822
HAL and Driver Documentation

EECS3215 Final Exam Option 2 - API Submission

Enclosed is my submission for EECS3215 Final Exam Option 2 for Winter 2019-2020. It is a HAL library for the Microchip MCP4822 12-bit dual-channel digital-to-analog converter. This DAC communicates over SPI and in calling the DAC initialization function the SPI bus is automatically initialized. This document gives a quick introduction, but much more detail on the library can be found in `MCP4822-api.pdf`.

This code has been developed on the LPC804 and while it should work on the LPC802, that has not been tested.

Included Files

- `readme.pdf` - this file
- `MCP4822.c` - source file containing driver code
- `MCP4822.h` - header file containing pin definitions, function prototypes, etc.
- `MCP4822-api.pdf` - detailed API documentation for defined functions, including usage examples
- `MCP4822-example.c` - minimum working example for the LPCXpresso804 development board. Flashing this to the board unchanged will result in a slow 2.048v amplitude sawtooth wave

Using the Library

To use this library in a project, copy `MCP4822.c` and `MCP4822.h` into the project directory and add the following line to the project file:

```
#include "MCP4822.h"
```

Modify the following pin definition lines so that the number in parentheses corresponds to the desired GPIO pins on the development board:

```
#define DAC0_SELECT   (20) // GPIO pin connected to CS pin on DAC
#define DAC0_LDAC     (2)  // GPIO pin connected to LDAC, if being used.
#define SPI_SCK_PIN   (16) // GPIO pin connected to SCK pin on DAC
#define SPI_MOSI_PIN  (17) // GPIO pin connected to MOSI (SDI) pin on DAC
#define SPI_SSEL0_PIN (1)  // GPIO pin for SSEL0. No connection to DAC needed.
```

Usage Examples

The SPI bus is initialized and the DAC is powered on to unity gain using the following code:

```
MCP4822_Init(DAC0_SELECT, GAIN_1);
```

A value of `4095` written to the DAC with unity gain will give an output of 2.048v (nominal). This can be done with the following line:

```
MCP4822_Write(DAC0_SELECT, DAC_CHANNEL_A, GAIN_1, POWER_UP, 4095);
```

Further usage examples and more details on configuration options are found in `MCP4822-api.pdf`