

# THE LPC84X MCU FAMILY – A ‘SWISS ARMY® KNIFE’ OFFERING OF FEATURES FOR YOUR NEXT IOT DESIGN

PART III  
KEVIN TOWNSEND  
BRENDON SLADE



PUBLIC



SECURE CONNECTIONS  
FOR A SMARTER WORLD

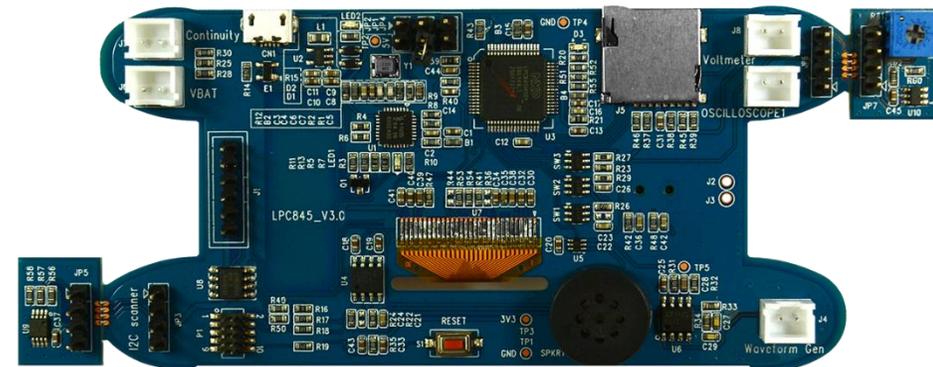
## Agenda – Part III



- Brief recap of the LPC84x multi-tester “Swiss army knife” for EEs (SAKEE) project
- Software development for SAKEE
  - Live demo of getting started with on LPC845 with MCUXpresso IDE
  - Other supported toolchains
  - Starting development with LPCXpresso boards
- Code overview

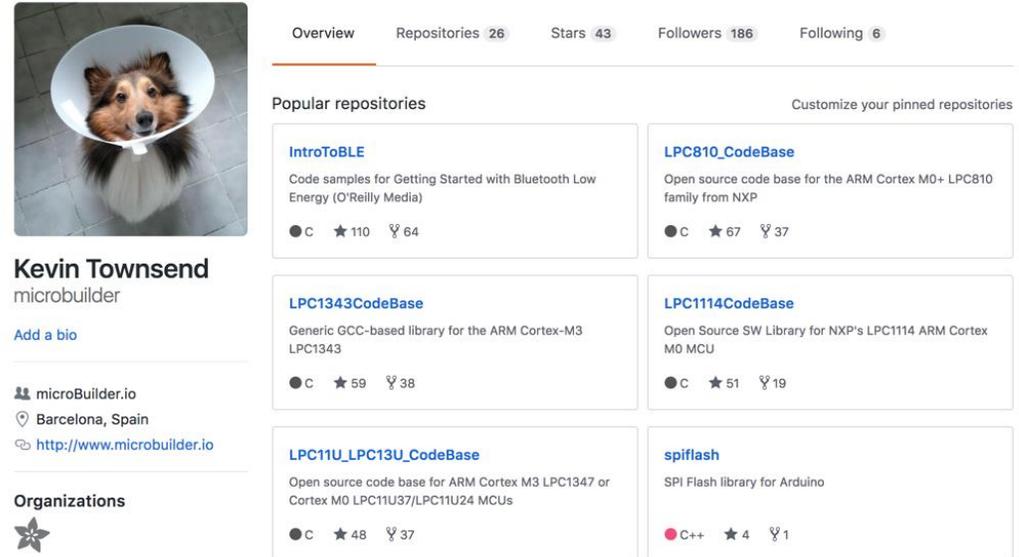
# Overview of Swiss Army Knife for EEs project

- Project aims
  - Develop a handy tool for basic testing tasks EEs frequently need to accomplish
  - Design a platform that can be easily customized and extended for specific testing tasks
  - Showcase how to use the main features of the highly flexible LPC845
  - Provide open source hardware and software to the LPC community of users for easy access to the design



# Kevin Townsend

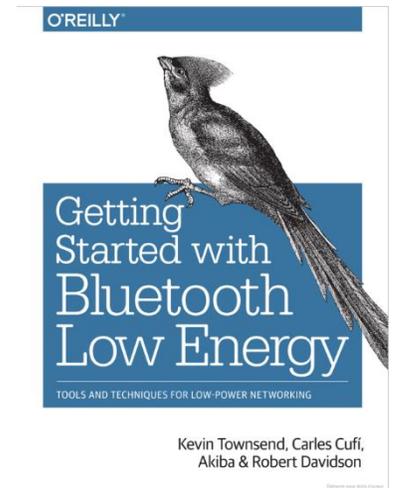
- Lead Engineer at Adafruit Industries
- Cortex-M user since the first commercially available M0 device (the LPC1114 in 2009)
- Published some of the earliest open source Cortex-M codebases for the LPC800, LPC1100, and LPC1300 families
- Many area of interest is extremely low cost 32-bit MCUs and RF solutions



Kevin Townsend's GitHub profile page. The profile picture shows a dog wearing a white funnel. The profile name is Kevin Townsend, with the handle @microbuilder. The bio includes the website microBuilder.io, location Barcelona, Spain, and the URL http://www.microbuilder.io. The profile shows 26 repositories, 43 stars, 186 followers, and 6 following. The 'Popular repositories' section lists: IntroToBLE (110 stars, 64 forks), LPC810\_CodeBase (67 stars, 37 forks), LPC1343CodeBase (59 stars, 38 forks), LPC1114CodeBase (51 stars, 19 forks), LPC11U\_LPC13U\_CodeBase (48 stars, 37 forks), and spiflash (4 stars, 1 fork).



[www.github.com/microbuilder](http://www.github.com/microbuilder)



# GETTING STARTED WITH MCUXPRESSO IDE



# MCUXpresso IDE - More Information

- General Product Information
  - [mcupresso.nxp.com](http://mcupresso.nxp.com)
- Where to download
  - [mcuxpresso.nxp.com](http://mcuxpresso.nxp.com)
- Docs
  - Within product : Built-in help system and PDFs
- Community support and FAQs
  - Monitored by full time, front line support team with worldwide timezone coverage
  - <https://community.nxp.com/community/mcuxpresso>

# MCUXPRESSO IDE LIVE DEMONSTRATION



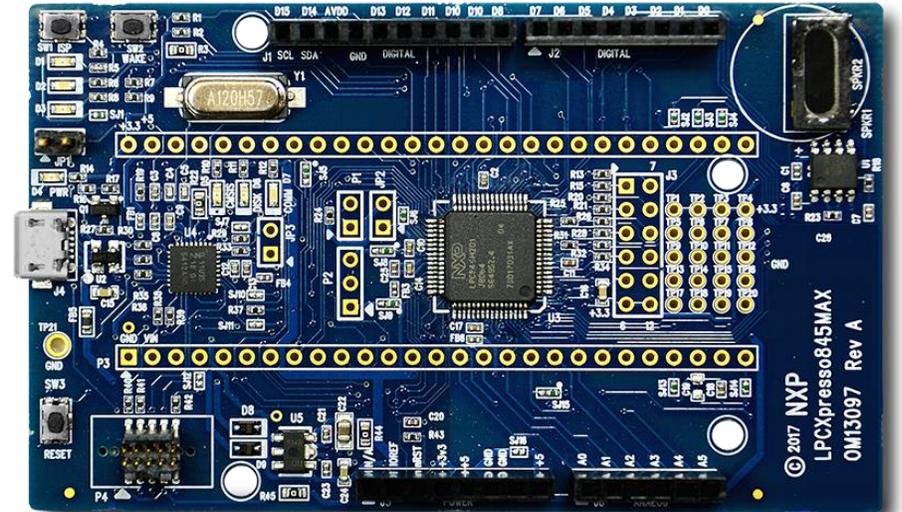
# LPCXpresso Boards

- Concept pioneered by NXP
  - Low cost, easy to use development platform with flexible expansion options
- Works with all partner tool-chains to provide a low-cost evaluation/development platform
- Common features
  - LPC Cortex-M processor with on-board debug probe
  - Connector for external debug probe
  - MAX, V2 and V3 boards offer popular Arduino R3 expansion
- Other features vary by target MCU



# LPCXpresso845MAX Board (OM13097)

- LPC845 in LQFP64 package
- *On-board CMSIS-DAP debug interface*
  - Supported by MCUXpresso IDE and popular 3rd party IDEs
  - Includes VCOM support (UART bridged via USB to host)
- Expansion options
  - Arduino UNO R3-compatible connectors
  - LPCXpresso and Pmod® options
  - Prototyping area
- User application test features
  - Red, Green and Blue user LEDs for test & debug
  - *On board speaker with driver*
  - User button
- Easily configured for LPC845 power measurement
- USB powered
- UART, SPI, I2C ISP boot-capable

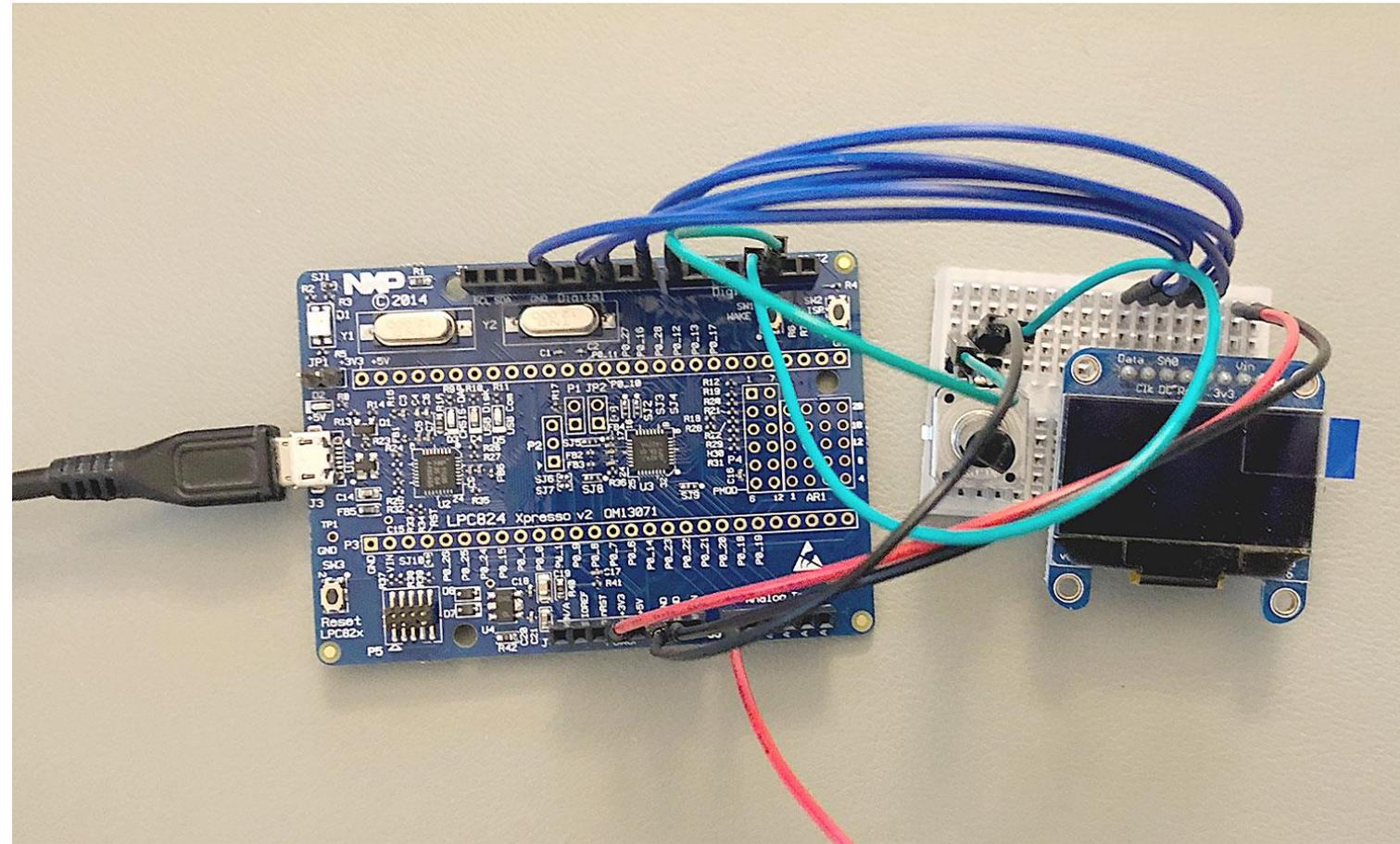


Orderable Part Number: **OM13097UL**  
<http://www.nxp.com/demoboard/OM13097>

*These features cloned on SAKEE*

# Early Prototypes

- Early development done on the LPC824-MAX (OM13071)
- High level of peripheral compatibility with LPC chips
- Easy and inexpensive to source
- Easy code and project migration when LPC845 available
- Developed using off the shelf parts from NXP and Adafruit Industries



# SOFTWARE DESIGN



# Codebase Summary

## ANALOG IO

- adc\_poll.c
- adc\_dma.c

## HMI (OLED/GFX, Buttons)

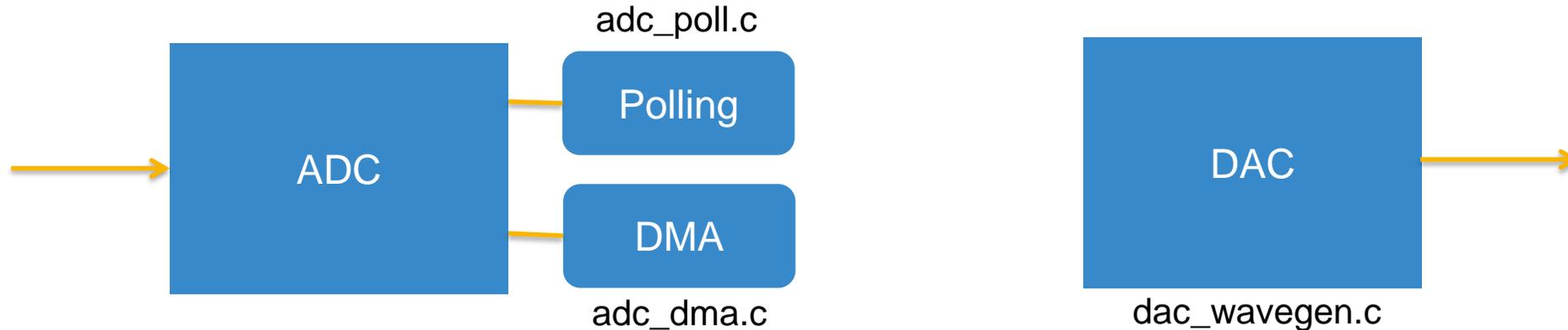
- ssd1306.c
  - gfx.c
- font5x7.h
- button.c
- qei.c
- qei\_sct.c

## APP MANAGEMENT

- app\_menu.c
- app\_about.c
- app\_cont.c
- app\_i2cscan.c
- app\_Scope.c
  - app\_vm.c
- app\_wavegen.c

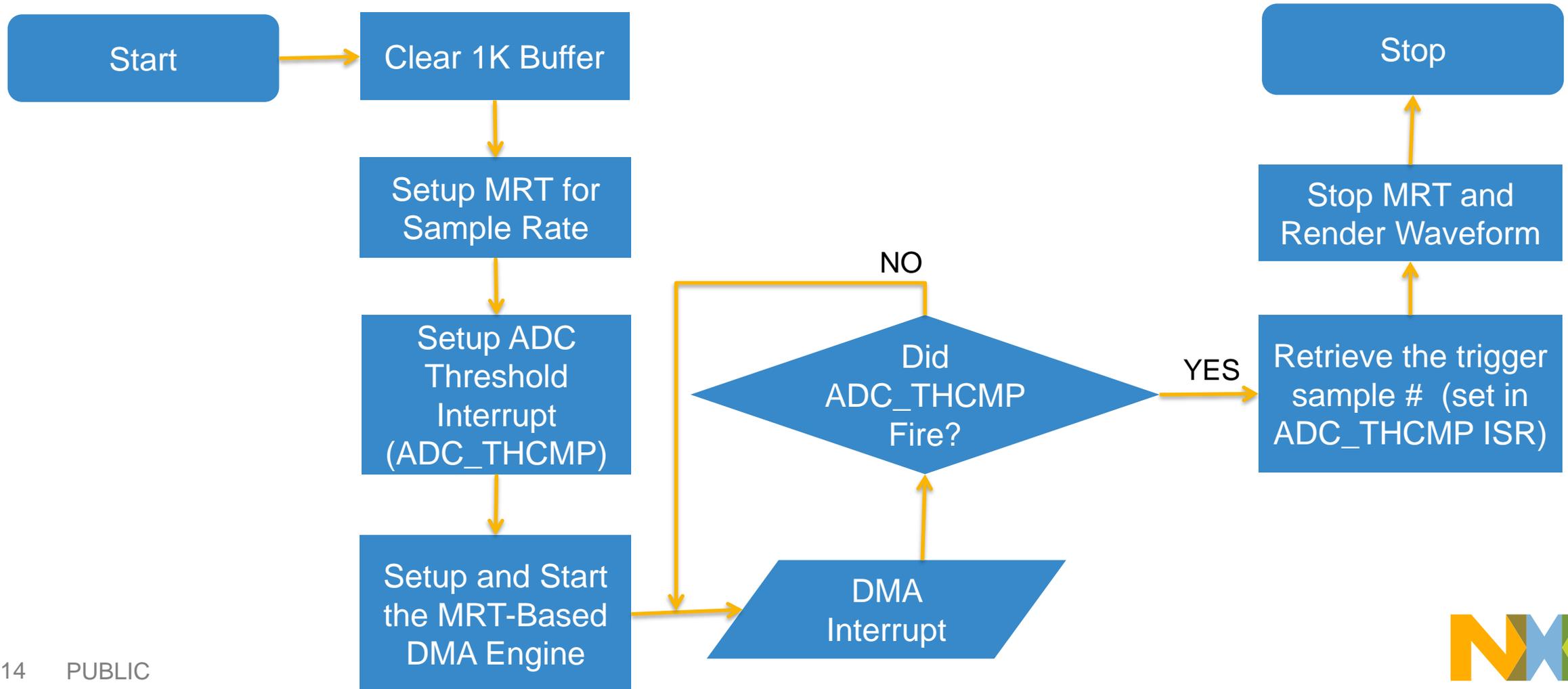
# Analog IO Components

- Both **polling** and **DMA-based** ADC drivers are present
  - The polling ADC code (`adc_poll.c`) is used by the voltmeter and for battery level, etc.
  - The DMA code (`adc_dma.c`) is used by the oscilloscope functionality
- The DAC driver (`dac_wavegen.c`) outputs analog waveforms based on lookup tables



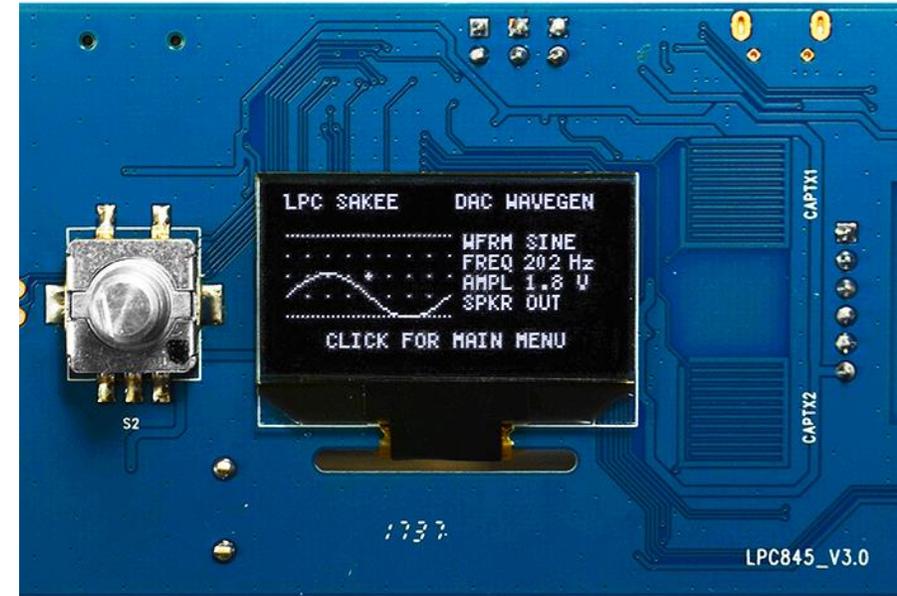
# Analog Input

- **adc\_poll.c**: Simply provides single ADC conversions on a specified channel
- **adc\_dma.c**: DMA-based driver using: MRT, DMA, 1K double-buffer, ADC Threshold IRQ



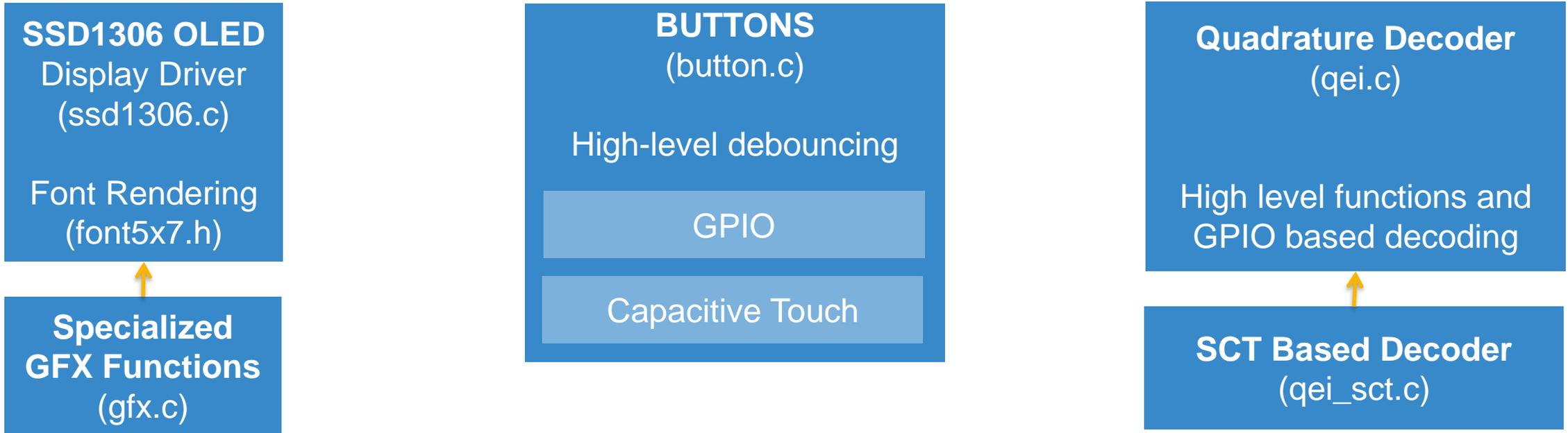
# Analog Output

- **dac\_wavegen.c** uses DAC1 to start/stop the analog waveform output
- Code is based on 64-sample 10-bit lookup tables
  - Can easily be extended if required
  - 64 samples was selected for easy rendering on the 128x64 pixel OLED display
- Variable rate waveform output
- Code can be easily reused for audio output, such as continuity buzzer
- Output filter reference design present, though this will need to be tailored to your output frequency



# HMI Components

- A number of reusable HMI components are present



# OLED Driver and GFX System

- **ssd1306.c** is an OLED driver based on the common SSD1306
- SPI Based (I2C support is relatively easy to add)
- Based on a 1K framebuffer (128\*64\*1b)
- Built-in text rendering (uses font5x7.h, easy to extend with other NXP fonts)
- Basic functions in code drivers:
  - Set pixel
  - Fill screen
  - Invert
  - Fill rectangle
- **gfx.c** contains specialized drawing functions to render:
  - Graticules (with configurable display properties)
  - 10-bit or 12-bit waveforms
  - Text rendering helpers to avoid sscanf and save control size

```
typedef enum
{
    GFX_GRATICULE_LINES_NONE = 0,           // No borders or divider lines
    GFX_GRATICULE_LINES_HOR = (1 << 0),    // Center horizontal line
    GFX_GRATICULE_LINES_VER = (1 << 1),    // Center vertical line
    GFX_GRATICULE_LINES_TOP = (1 << 2),    // Top border
    GFX_GRATICULE_LINES_BOT = (1 << 3),    // Bottom border
} gfx_graticule_lines_t;

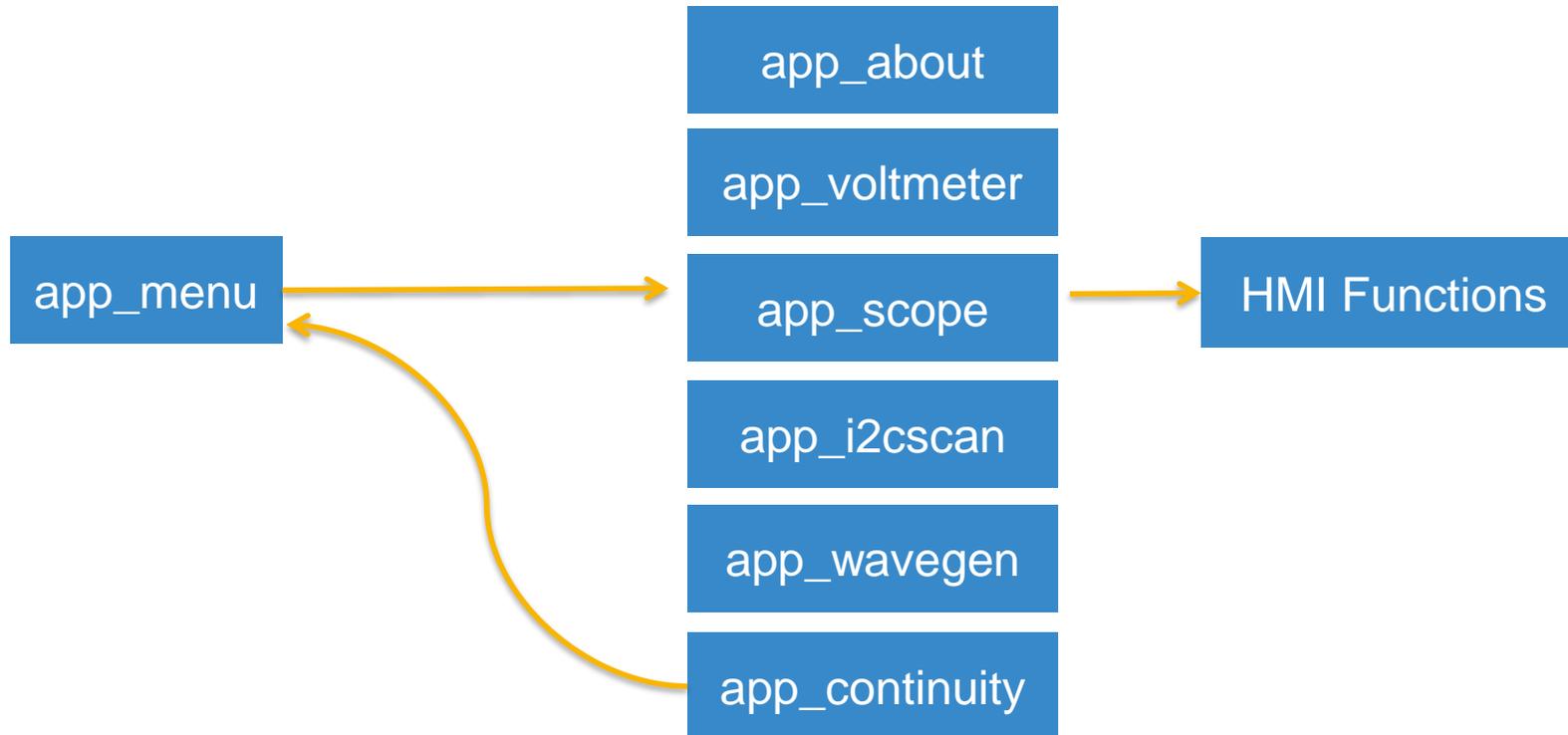
typedef struct
{
    uint8_t w;
    uint8_t h;
    gfx_graticule_lines_t lines;
    uint8_t line_spacing;                  // Space between each dot on the hor/ver/etc dividers
    uint8_t block_spacing;                // Pixels in each divider in the graticule
} gfx_graticule_cfg_t;

int gfx_bar(uint8_t x, uint8_t y, uint8_t color, uint8_t height);
int gfx_waveform_64_32(uint8_t x, uint8_t y, uint8_t color, const uint16_t *wform, int16_t offset, uint16_t bufsize, uint8_t rshift, uint8_t rshift_mask);
int gfx_waveform_64_32_10bit(uint8_t x, uint8_t y, uint8_t color, const uint16_t *wform, int16_t offset, uint16_t bufsize, uint8_t rshift, uint8_t rshift_mask);
int gfx_graticule(uint8_t x, uint8_t y, gfx_graticule_cfg_t *cfg, uint8_t color);
int gfx_printhex8(uint8_t x, uint8_t y, uint8_t hex, uint8_t scale, uint8_t color);
int gfx_printdec(uint8_t x, uint8_t y, int32_t dec, uint8_t scale, uint8_t color);

uint8_t gfx_num_digits(uint32_t x);
```

# App Management

- Each 'application' has it's own \*\_init and \*\_run function
- **App\_menu** is used as the parent app
  - Executes inside the super-loop
  - Calls sub-applications, which return when complete
  - HMI code is shared throughout sub-applications



```
// Run the appropriate sub-app
switch (option)
{
case APP_MENU_OPTION_LAST:
case APP_MENU_OPTION_ABOUT:
    // Display 'About' menu
    app_about_init();
    app_about_run();
    break;
case APP_MENU_OPTION_VOLTMETER:
    // Init voltmeter
    app_vm_init();
    app_vm_run();
    break;
case APP_MENU_OPTION_SCOPE:
    // Init scope
    app_scope_init();
    app_scope_run();
    break;
case APP_MENU_OPTION_I2CSCANNER:
    // Init I2C bus scanner
    app_i2cscan_init();
    app_i2cscan_run();
    break;
case APP_MENU_OPTION_WAVEGEN:
    // Init wavegen
    app_wavegen_init();
    app_wavegen_run();
    break;
case APP_MENU_OPTION_CONTINUITY:
    // Init continuity tester
    app_cont_init();
    app_cont_run();
    break;
}
```

## Where to find out more

- Visit Developer Resources > Reference Designs > LPC845 Based Swiss Army Knife Multi-tester
  - Direct URL: <http://www.nxp.com/pages/:LPC845-Multi-Tester>
- Kevin's Github (software) : [www.github.com/microbuilder/LPC84x\\_SAKEE](http://www.github.com/microbuilder/LPC84x_SAKEE)
- More information on LPC84x: <https://www.nxp.com/lpc84x>
- MCUXpresso IDE: <https://www.nxp.com/mcuxpresso>

**Thanks for watching!**



**SECURE CONNECTIONS  
FOR A SMARTER WORLD**