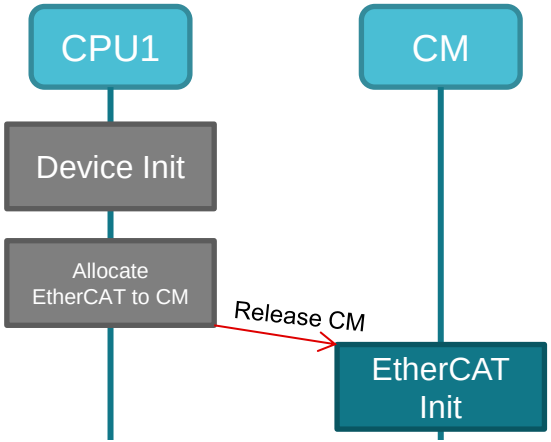
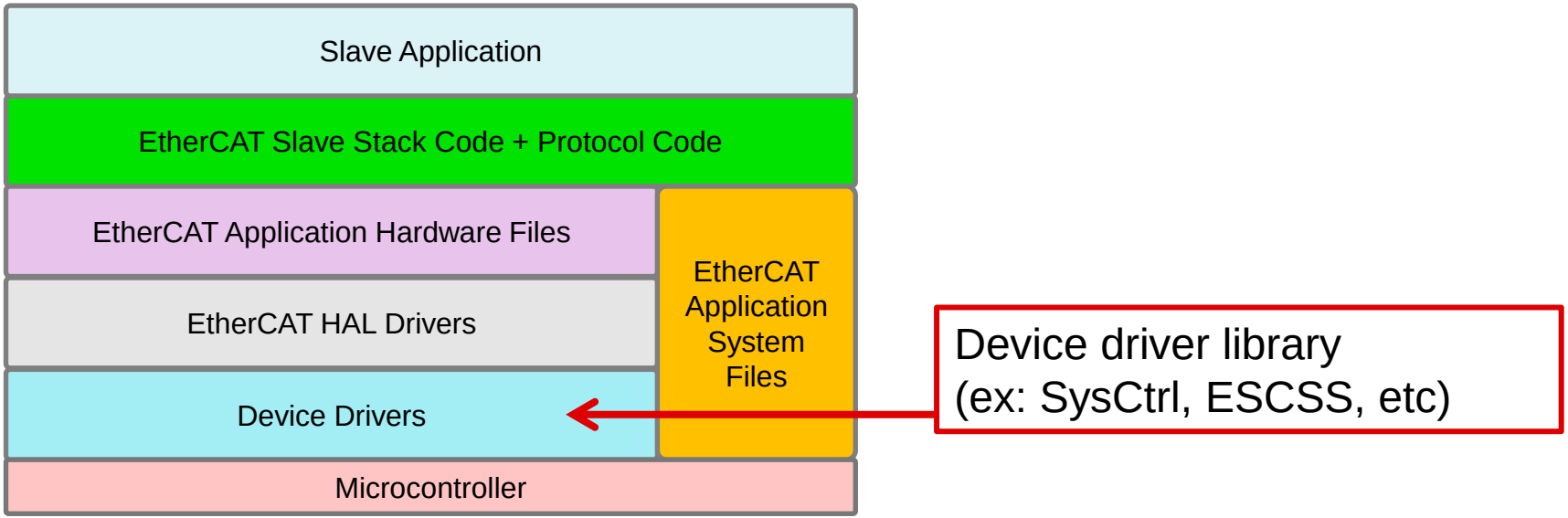
EtherCAT F2838x Application Overview

General F2838x Application Flow (Example)

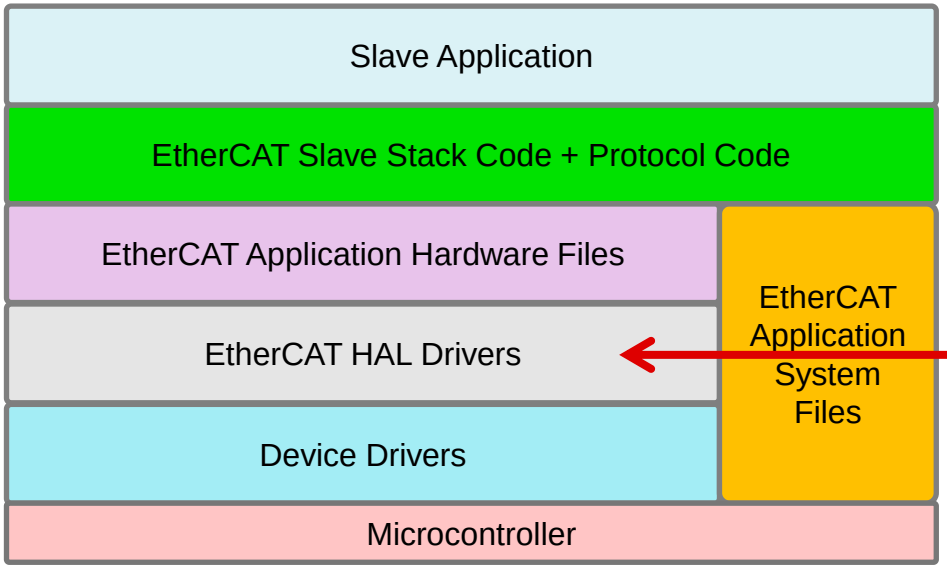
- **App Initialization**
  - CPU1 application owns:
    - Device initialization
    - Allocating EtherCAT to CM and releasing CM from reset
  - CM application owns:
    - EtherCAT IP and Stack initialization
- **App Main Loop**
  - CPU1 application owns:
    - Running the motor control algorithm
  - CM application owns:
    - Running the EtherCAT slave stack
    - Transmitting EtherCAT data via IPC to CPU1 and receiving EtherCAT data from CPU1 to transmit to EtherCAT master



F2838x ESC Software Stack



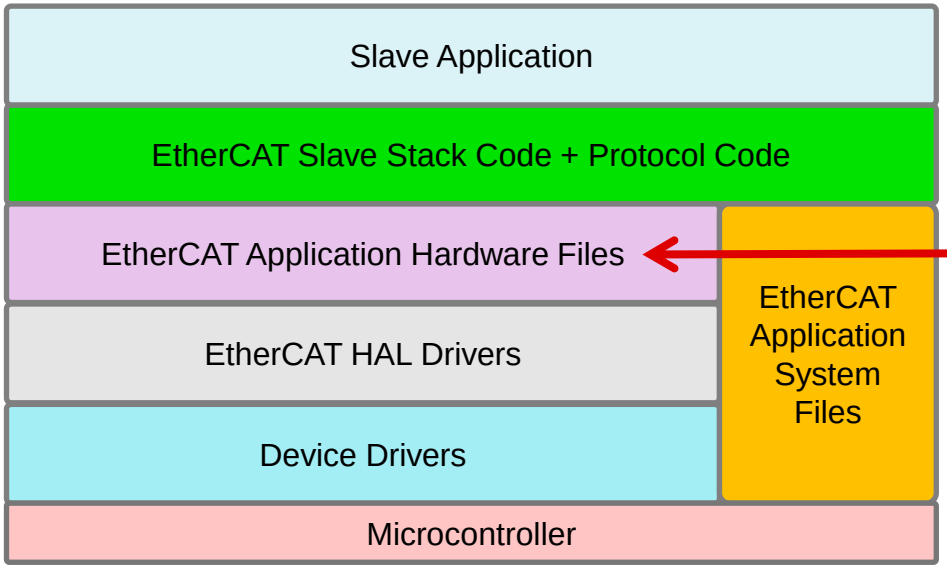
F2838x ESC Software Stack



Device CPU-specific HAL drivers

- Init the EtherCAT subsystem
- APIs for EtherCAT read/write

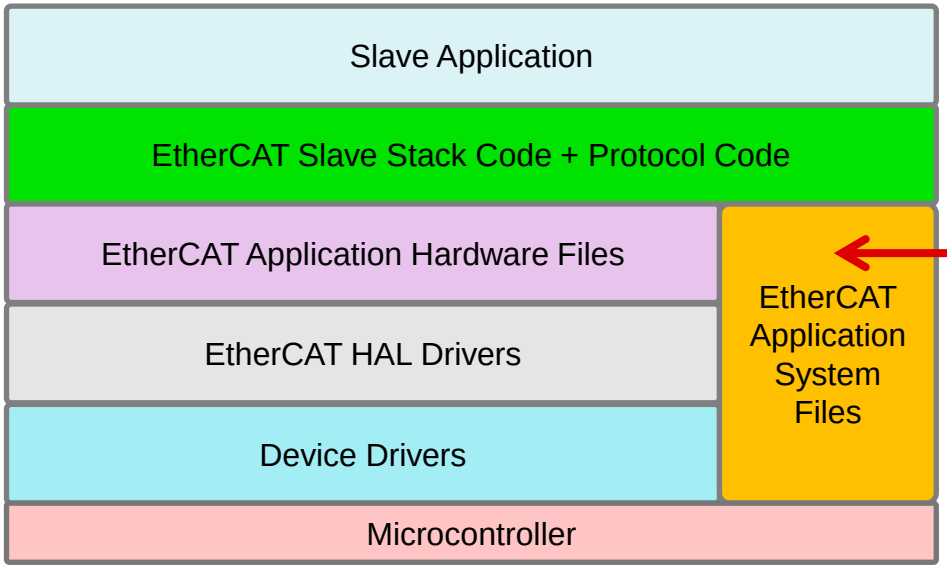
F2838x ESC Software Stack



Device CPU-specific hardware header file

- Maps HAL driver functions to stack API names

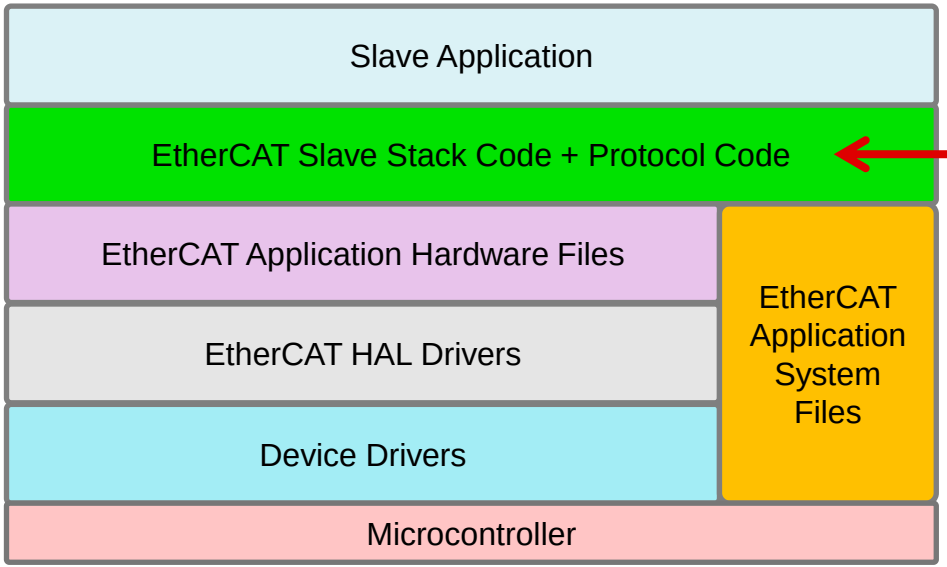
F2838x ESC Software Stack



Device CPU-specific application system source/header file

- Wrapper functions for memcpy, memset, etc
- Primarily needed for C28x to handle byte input sizes

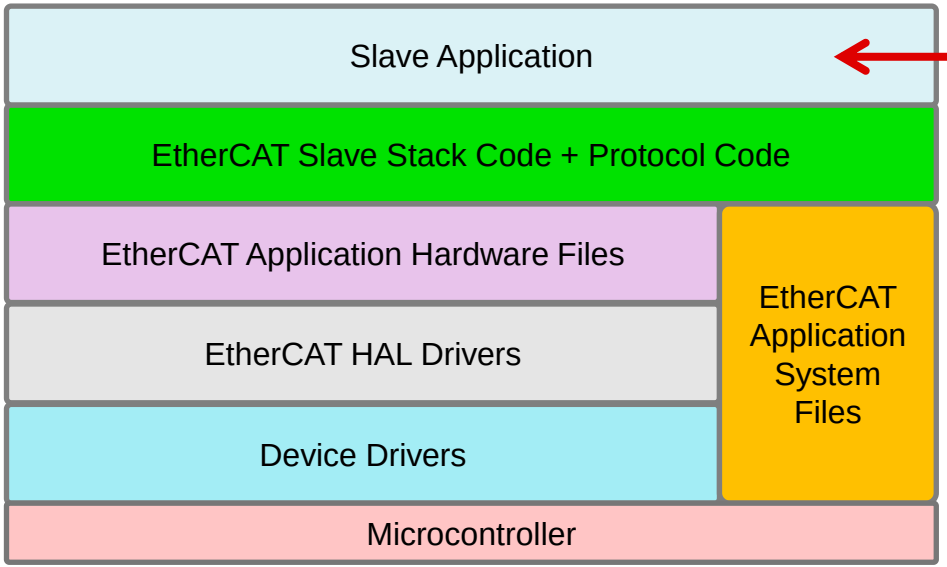
F2838x ESC Software Stack



Slave stack code source/header files generated from the SSC tool.

- Main slave state machine
- Contains protocol code (ex: CAN over EtherCAT)

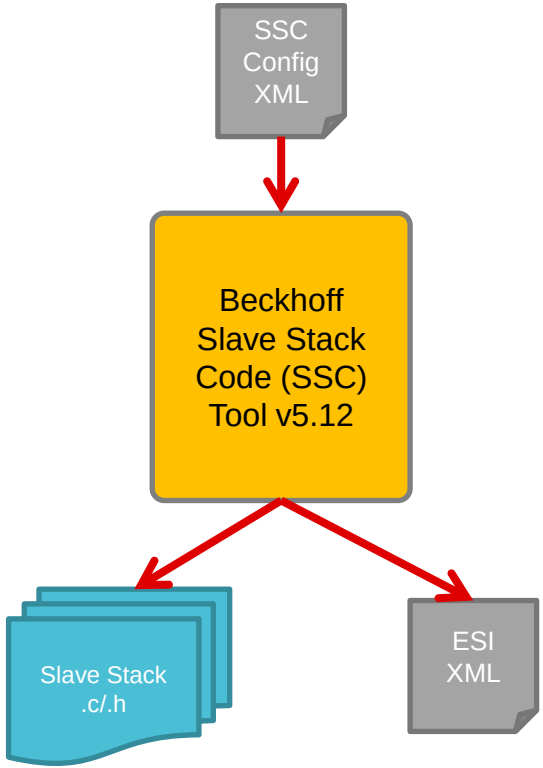
F2838x ESC Software Stack



User slave application
- Slave main loop
- Defines API handlers required by stack (ex: called during state changes)
- Defines object dictionary and/or input/output data variables

Generating Stack Code for F2838x

1. Download C2000Ware
2. Locate F2838x EtherCAT collateral
   - ~\libraries\communications\Ethercat\f2838x
3. Run the EXE to extract necessary files
4. Download and run Beckhoff Slave Stack Code Tool
   - Must be v5.12
5. Import the SSC configuration XML (extracted earlier) into the SSC tool
6. The SSC drop down menu will display 4 options
   - 2 options that include an example application with stack code
   - 2 options that include slave stack code only
7. SSC generates the stack source code and ESI XML



Note: This flow is described in more detail in the EtherCAT SW Guide in C2000Ware

Getting Started with EtherCAT on F2838x

Resource	Description	Location
F28388D controlCARD	• F28388D controlCARD evaluation module • Includes 2-port EtherCAT PHY connections	www.ti.com/tool/TMDSCNCD28388D
F2838x TRM – EtherCAT Chapter	• ESC subsystem integration features • ESC functional blocks • ESC physical layer • ESC interfaces to CPU1/DMA/CLB/other peripherals • ESC subsystem registers	www.ti.com/product/TMS320F28388D
EtherCAT SW Guide	• EtherCAT App and Stack Software Overview • Instructions on using the CPU1/CM EtherCAT examples • How-To procedures on setting up TwinCAT (or EC-Engineer), programming EEPROM, generating Slave Stack Code, etc • Common example troubleshooting • Details on CPU1/CM HAL APIs	www.ti.com/tool/C2000WARE (~\libraries\communications\Ethercat\f2838x)
EtherCAT CPU1/CM Stack Generation and Config Files	• Slave Stack Code Tool Configuration file - Used to import into SSC tool and generate stack configured for Tenor • Required hardware, system, and HAL source/header files for using the EtherCAT slave stack	www.ti.com/tool/C2000WARE (~\libraries\communications\Ethercat\f2838x)

Getting Started with EtherCAT on F2838x

Resource	Description	Location
EtherCAT CPU1/CM Echoback examples	• Demonstrate basic master to slave to master communication • Data of varying sizes (8 bit, 16-bit, 32-bit) is send from master to slave and looped back by slave to master	www.ti.com/tool/C2000WARE (~\libraries\communications\Ethercat\f2838x)

EtherCAT Connected FCL Sensored PMSM IDDK Example	• Demonstrate controlling speed/position of motor via EtherCAT
	• CPU1 is running motor control loop and CM is running EtherCAT slave stack
	• EtherCAT master (TwinCAT) sends speed/position command and speed/position reference data to slave.
	  ◦ CM sends this data via IPC to CPU1 and CPU1 responds with speed/position/torque/drive statuses
	  ◦ CM (EtherCAT slave) sends this status information back to the EtherCAT master
		www.ti.com/tool/C2000WARE-MOTORCONTROL-SDK
		(~\solutions\tmdxiddk379d\f2838x)