

MCS C Development Platform

 **BOSCH** GTM Tech Day 2024



Saarbrücken Germany, 08.03.2024



Contribute to a secure and safer world with highest reputation and market share.



About HighTec



- An **independent**, privately owned company, since 1982
- **Worldwide** presence, offices and partners



- Specialized on embedded development **tools** and **design-in services**

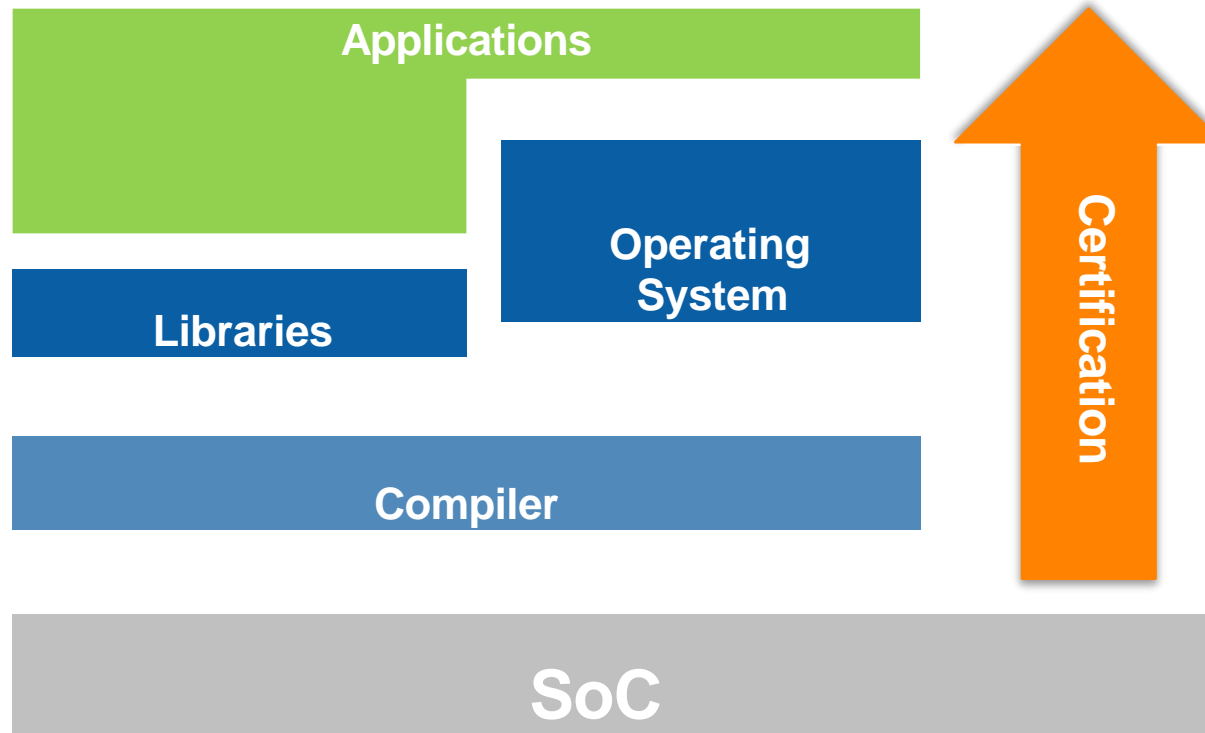


ISO 26262 ASIL D



- Focus on **automotive** and **industrial** markets with high safety demands

Certified From Bottom to the Top



ISO 26262:2018 - Cars, motorcycles, trucks, buses, trailers and semi-trailers

IEC 61508:2010 – Electrical / electronic / programmable electronic safety-related systems

HighTec – Safety Certificates



Addressing the increased demand for safety applications



- Software
- C-Library
- ISO 26262:2018



- Software
- Operating System
- IEC 61508 (SIL3)
- ISO 26262 (ASIL D)



- Functional Safety
- IEC 61508



- Functional Safety
- ISO 26262



- ISO 9001

Multi-Architecture Compiler

Automotive
Grade Compilers

ISO 26262 ASIL D
Qualification



Long Term
Support

40 Years of
Safety Expertise

32-bit Compiler



TC4x

TC3x

TC2x



S32

Z	E
G	K
R	S



64-bit Compiler

arm



HPC



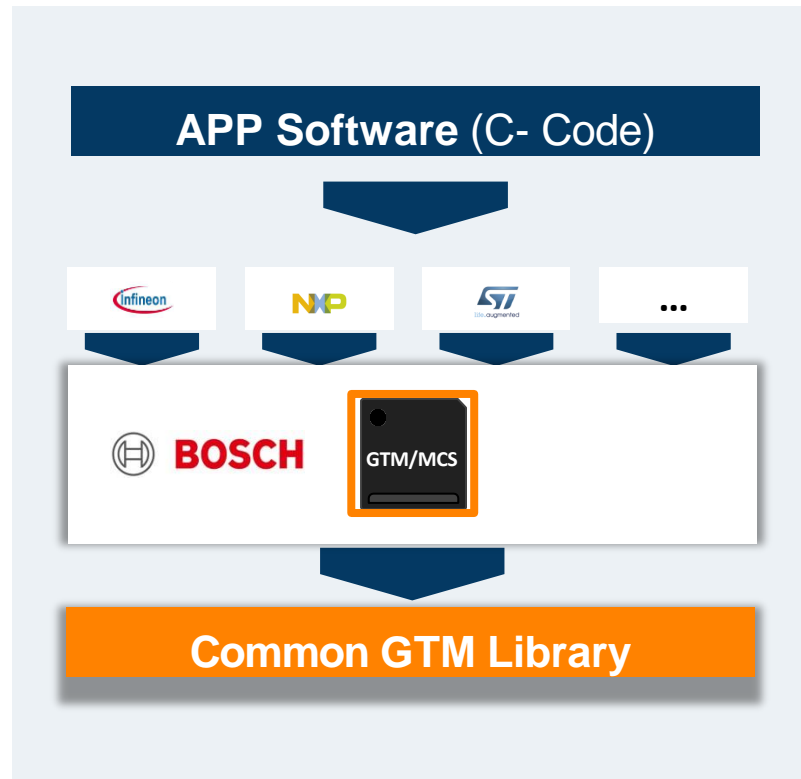
No.1 Automotive Grade LLVM Compiler



HighTec makes innovative open-source technology automotive grade for silicon vendors

GTM – Vision – Portability in mind

Unified complex timer functions across silicon vendors



- ❑ GTM/MCS **minimizes interdependencies** between timer software functions
- ❑ Ensure **real-time** requirements
- ❑ MCS can server **multiple IRQs**
- ❑ Host can change **GTM configuration**
- ❑ **Challenges:** Low level real-time programming - Heterogenous multi-core - Safety - Ecosystem - Technical support

GTM/MCS C Compiler

Overview



**Latest
GTM-IP v4.x
Support**

**ANSI/ISO
C99
GTM EABI**

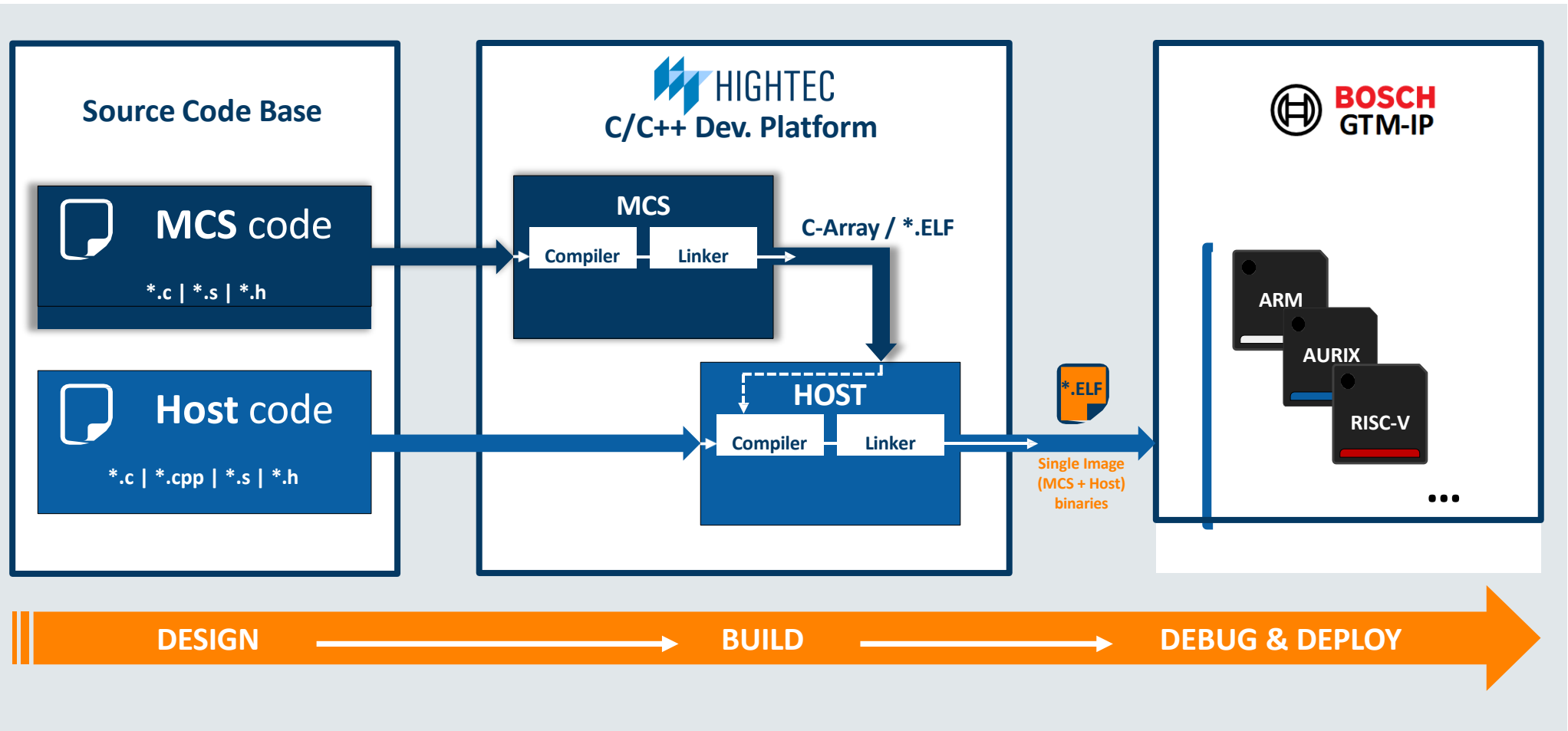
**Built-in
Intrinsic
Functions**

SFR, WAIT, ARU/AEI

**3rd Party
Debugger
Support**

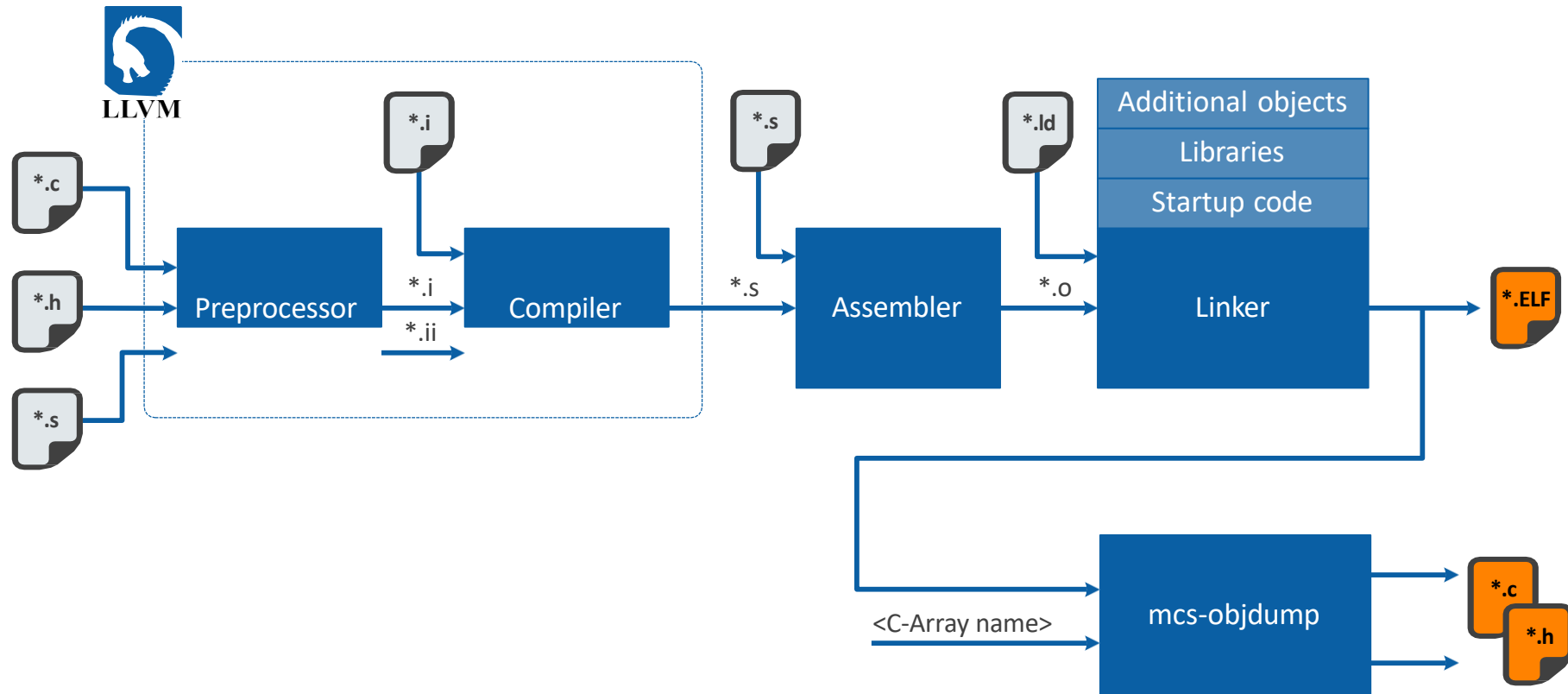
HighTec's Unified Development

GTM/MCS in the heterogeneous multicore environment



MCS Source Code Build Process

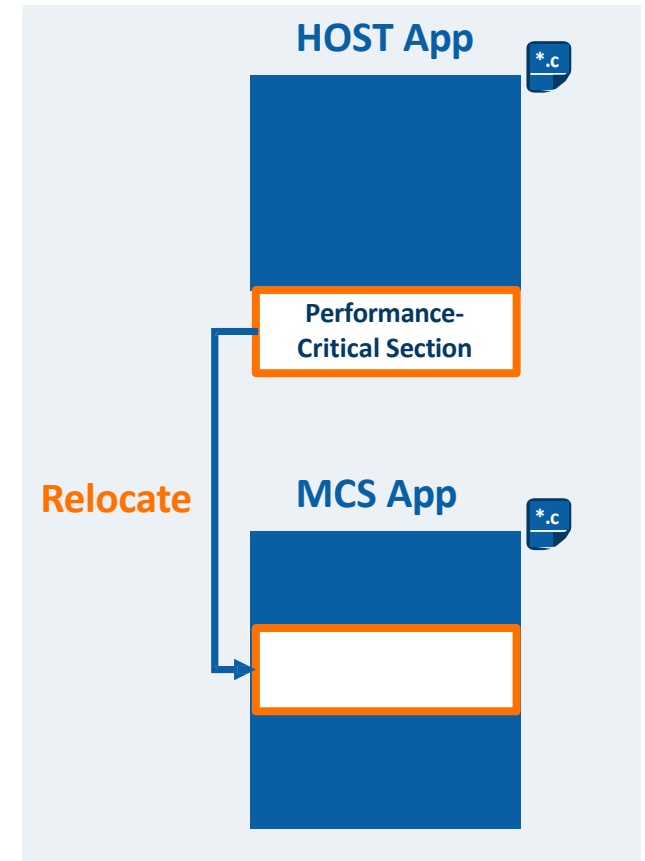
MCS Development Suite



Relocating Application Code to MCS

Offloading Code to MCS for Enhanced Performance

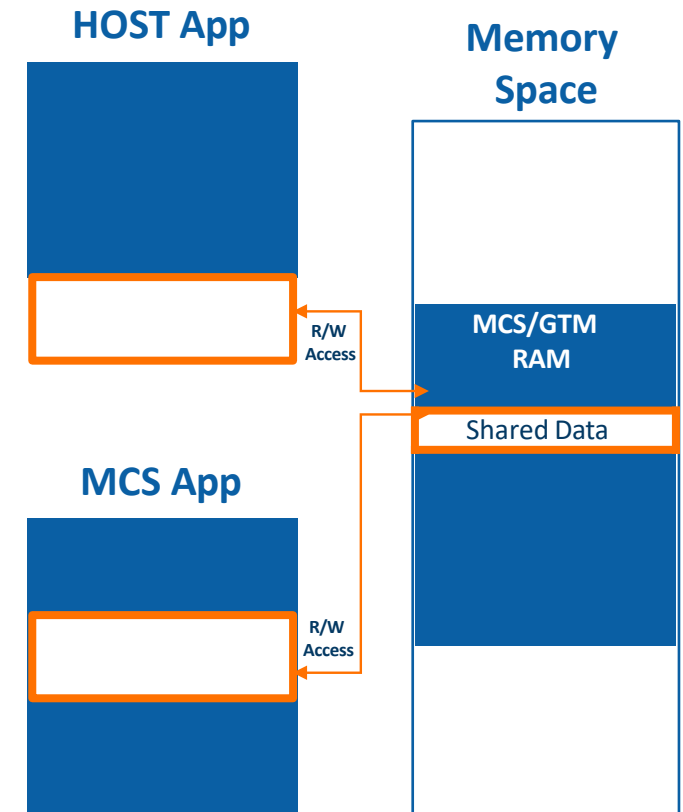
- **Objective of Relocation:**
 - Improve execution performance by offloading critical sections of the application, such as regulation loops in BLDC motor control, from the host core to the MCS.
- **Relocation Process:**
 - Identify performance-critical sections of code running on the host.
 - Relocate these sections to the MCS project, which is compiled separately for optimized execution.
- **Execution Strategy:**
 - Code can be executed on a single MCS channel or divided across multiple channels for sequential execution.
 - Execution occurs directly from GTM-RAM.
- **Initialization and Control:**
 - GTM-RAM memory initialization is managed by the host core, setting up environment for MCS execution.
 - MCS can handle interrupts and accessing peripherals, facilitating seamless operation within the system.



Data Sharing Between Host and MCS

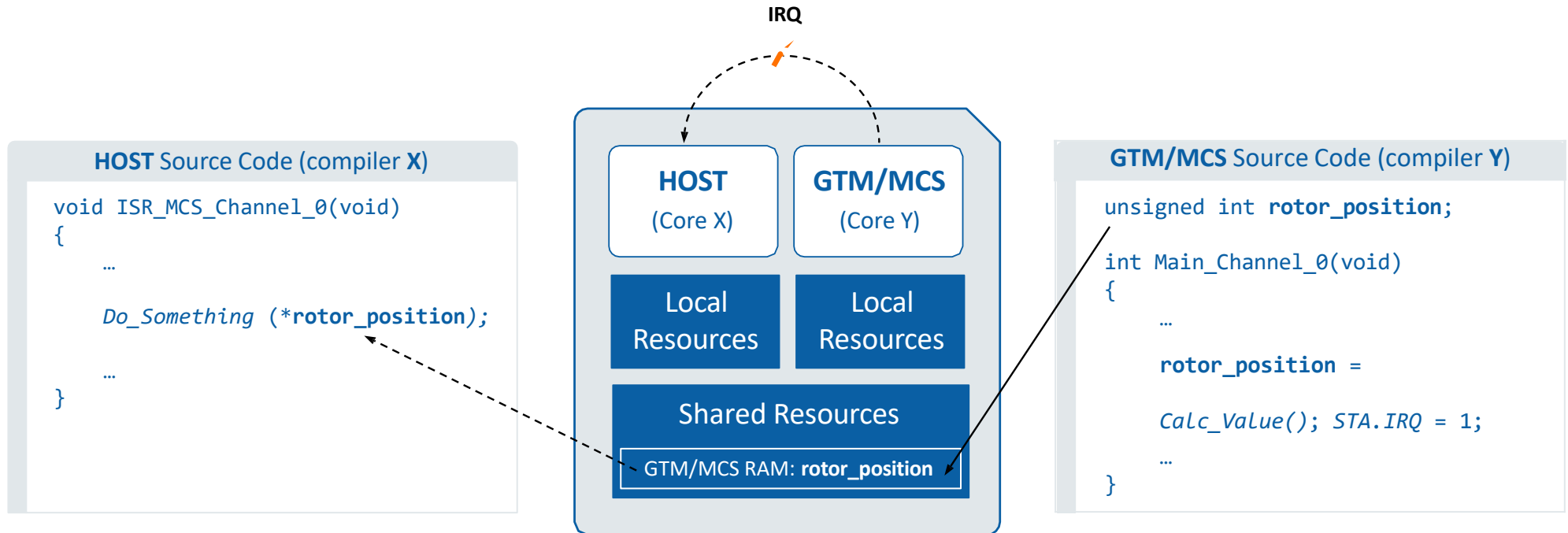
Efficient Data Sharing in a Heterogeneous Core System

- **Location of Shared Variables:**
 - Shared variables and common data must be located in MCS/GTM-RAM for accessibility by both the host core and MCS.
- **Accessing Shared Data:**
 - Ensures both the host application and MCS can efficiently access and modify shared variables.
 - Facilitates real-time data exchange and synchronization between the Host and MCS.
- **Awareness and Considerations:**
 - The heterogeneous nature of the core system requires specific handling and awareness of the distinct operational domains.
 - Developers must be aware of the challenges and advantages of operating in a mixed-core environment, optimizing code and data structures accordingly.



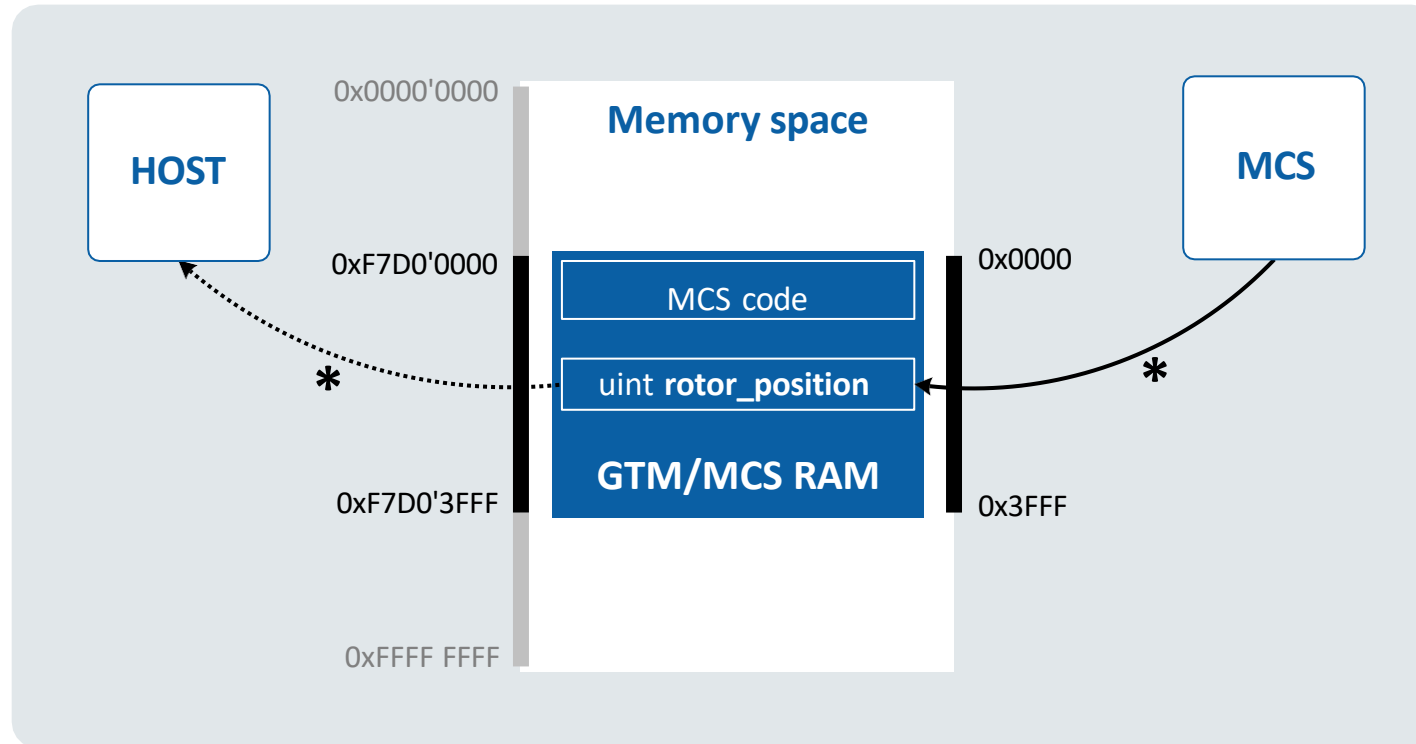
Example

Exporting MCS variables to the Host code



GTM/MCS RAM Access

MCS versus Host access to the MCS RAM



&rotor_position = 0xF7D0'03FC

&rotor_position = 0x03FC

MCS Code Image as a C-Array

The files become part of the **Host** application source code

mcs00.c

*.c

```
#include "mcs00.h"

unsigned long mcs00[263] = {
/* Section: .mcs_start */
/* _start [0]: */
    0xE0000020,
    0xE0000054,
    0xE0000088,
    0xE00000BC,
    0xE00000F0,
    0xE0000124,
    0xE0000158,
    0xE000018C,
/* mcs_start_ch0 [8]:
   */ 0xA001041C,
    0x20000001,
    0xA002041C,
    ...
};
```

mcs00.h

*.h

```
#ifndef mcs00_h_
#define mcs00_h_

extern unsigned long mcs00[263];

#define OFFSET_mcs00                (0)
#define SIZE_mcs00                  (1052)

#define LABEL_mcs00_mcs_start_ch0   (8)
#define LABEL_mcs00_mcs_exit_ch0    (20)
#define LABEL_mcs00_mcs_main_ch0    (131)
...

#define LABEL_mcs00_rotor_position   (255)
#define LABEL_mcs00_SYSTEM_STACK_CH0 (271)

...
#endif
```


C-Array - Integration

Handling of the MCS image in the Host application code

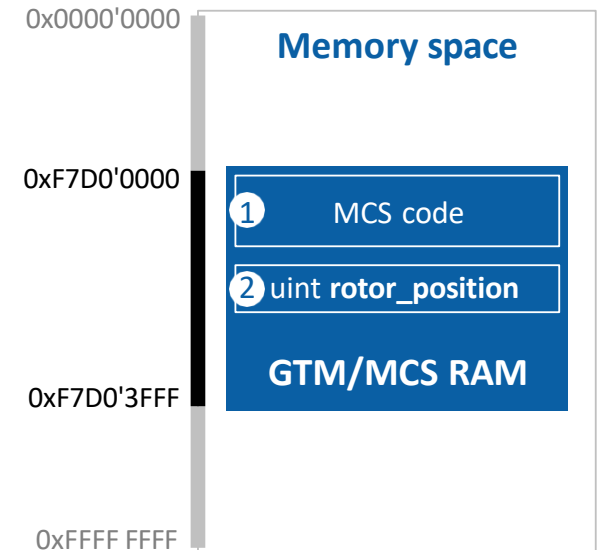
1 Copy the MCS image to the MCS RAM

Include the `mcs00.h`

Copy the unsigned long `mcs00[...]`

2 Declare links to the MCS export variables

```
uint *rotor_position = (uint*) (0xF7D00000 +  
                                (LABEL_mcs00_rotor_position * 4));
```



HighTec – 5 Facts

Addressing the increased demand for Safety and Security



- 1 **No.1 Open Source compiler with **Safety (ASIL D)****
- 2 **Key software provider for OEMs and Tier1s**
- 3 **Worldwide presence: EMEA, APAC and US and partner network to develop custom solutions**
- 4 **Close cooperation with **semiconductor** vendors**
- 5 **Supporting material for all major automotive uC**



5 Min. Q&A Session



www.hightec-rt.com



support@hightec-rt.cn



sales@hightec-rt.cn