```
                        &shmem_attributes,
                        sizeof(shmem_attributes),
                        &status);

         if (status != MRAPI_SUCCESS) { ERR(“Unable to create shmem”);}
         break;
      }
   }
}
```

6.2 Automotive Use Case

6.2.1 Characteristics

6.2.1.1 Sensors

Tens to hundreds of sensor inputs read on periodic basis. Each sensor is read and its data are processed by a scheduled task.

6.2.1.2 Control Task

A control task takes sensor values and computes values to apply to various actuators in the engine.

6.2.1.3 Lost Data

Lost data is not desirable, but old data quickly becomes irrelevant, most recent sample is most important.

6.2.1.4 Types of Tasks

Consists of both control and signal processing, especially FFT.

6.2.1.5 Load Balance

The load balance changes as engine speed increases. The frequency at which the control task must be run is determined by the RPM of the engine.

6.2.1.6 Message Size and Frequency

Messages are expected to be small and message frequency is high.

6.2.1.7 Synchronization

Synchronization between control and data tasks should be minimal to avoid negative impacts on latency of the control task; if shared memory is used there can be multiple tasks writing and one reader; deadlock will not occur but old data may be used if an update is not ready.

6.2.1.8 Shared Memory

Typical engine controllers incorporate on-chip flash and SRAM and can access off-chip memory as well. Shared memory regions must be in the SRAM for maximum performance. Because small OS or no OS is involved it is typical for logical mappings of addresses to be avoided, where an MMU is involved it will typically be programmed for logical == physical and with few large page entries versus lots of small page entries. Maintenance of a page table and use of page replacement algorithms is to be avoided.