@odinthenerd

– not the god

ARM CMSIS

odinthenerd commented on 25 Jul 2016 • edited ▾ Contributor

If I am interpreting this correctly code resulting in the following (psydo) assembler would allow the load to happen before the pending ISR and thus causing many bugs in mbed.

```
str r0, ICER   #store mask into ICER (also known as NVIC_CLRENA on other chips)
ldr r1, r2       #load from SFR
<---- interrupt happens here because clear has not propagated yet
#use r1 although it is outdated due to modifications in the ISR
```

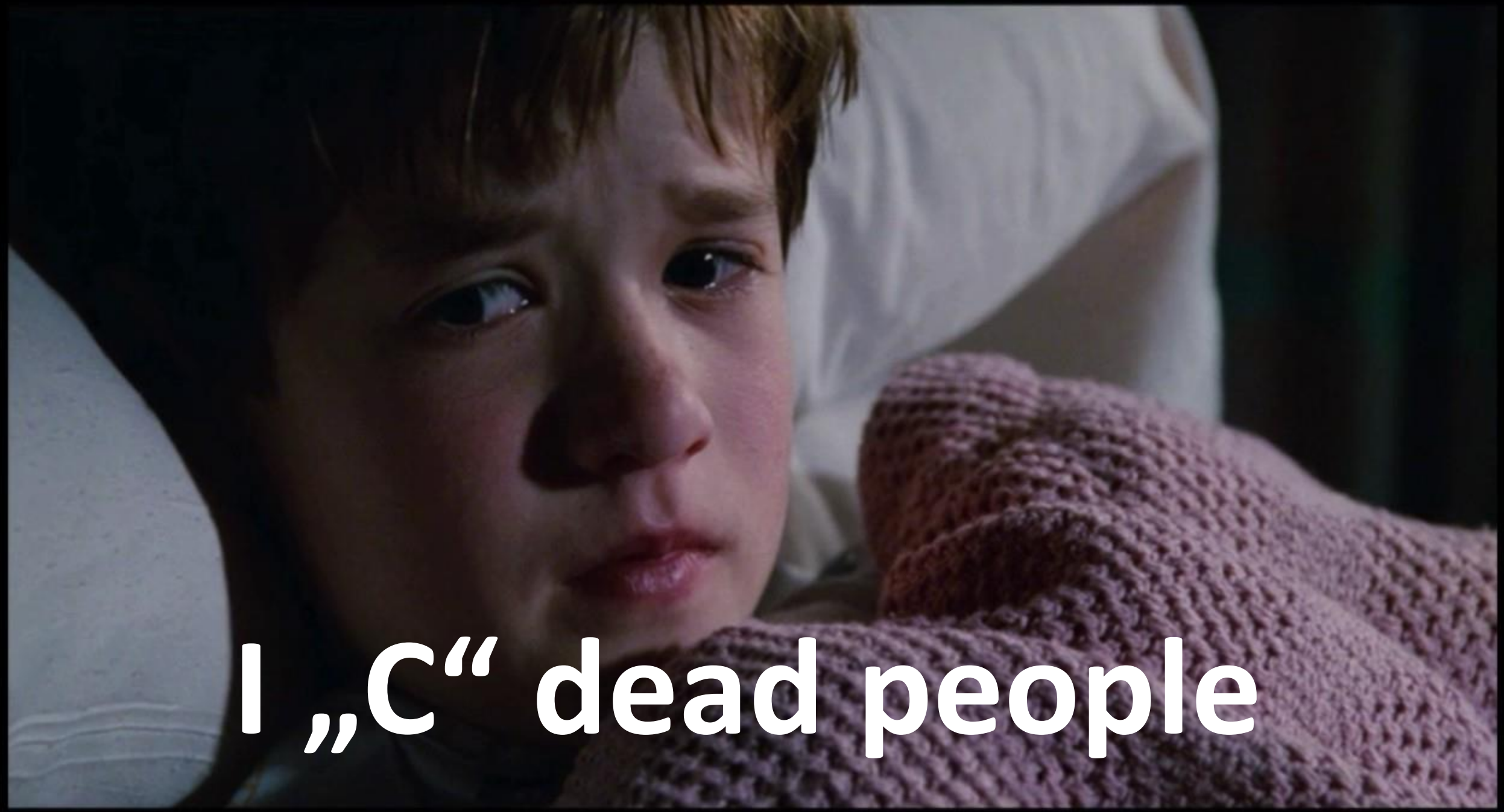
this would be one such example of a possible bug (after optimization of course)

Atomic queue

```
ring[p_writer++] = new_data;



//other thread
if(p_reader != p_writer)
    use(ring[p_reader++]);
```

Atomic queue

```
ring[p_writer++] = new_data;
if(p_writer == std::end(ring))
    p_writer = ring;

//other thread
while(p_reader != p_writer){
    use(ring[p_reader++]);
    if(p_reader == std::end(ring))
        p_reader = ring;
}
```


I „C“ dead people

Atomic queue

```
ring[p_writer++] = new_data;
if(p_writer == std::end(ring))
    p_writer = ring;

//other thread
while(p_reader != p_writer){
    use(ring[p_reader++]);
    if(p_reader == std::end(ring))
        p_reader = ring;
}
```

Lock?

```
HAL_StatusTypeDef HAL_ADC_Start_DMA(ADC_HandleTypeDef* hadc,
            uint32_t* pData, uint32_t Length)
{
  __IO uint32_t counter = 0;

  /* Check the parameters */
  assert_param(IS_FUNCTIONAL_STATE(hadc->Init.ContinuousConvMode));
  assert_param(IS_ADC_EXT_TRIG_EDGE(hadc->Init.ExternalTrigConvEdge));

  /* Process locked */
  __HAL_LOCK(hadc);

//…
```

Throw lock!

```
#define __HAL_LOCK(__HANDLE__)                                  \
                                do{                              \
                                    if((__HANDLE__)->Lock == HAL_LOCKED)   \
                                    {                            \
                                       return HAL_BUSY;          \
                                    }                            \
                                    else                         \
                                    {                            \
                                       (__HANDLE__)->Lock = HAL_LOCKED;    \
                                    }                            \
                                  }while (0U)
```

Throw lock!

```
/* Start the DMA channel */
HAL_DMA_Start_IT(hadc->DMA_Handle, (uint32_t)&hadc->Instance->DR,
    (uint32_t)pData, Length);
```

@odinthenerd

- Github.com
- Twitter.com
- Gmail.com
- Blogspot.com
- LinkedIn.com
- Embo.io