IEEE 1722 GPIO

David Ortiz

IEEE 1722b Task Force – 21st June 2022

Introduction & objectives I

- Ongoing work at IEEE 1722b already covers key ACF protocols that are needed to support high speed sensor connections on the car network.
  - For example, the generic sensor interface for the high-speed data or GBB protocol and I2C protocols for the sensor control.
- For generic sensor control it is commonly required to control several GPIO lines on the sensor:
  - Control of the Reset line, to control boot process, or to recover from errors.
  - Detection of interrupts from sensor (this is common on Radar MMIC’s).
  - Generation of frame synchronization signal on cameras.

Introduction & objectives II

- It could be possible to support this using/adapting existing ACF protocols:
  - Control of GPIOs outputs could be done using currently defined ACF protocols by memory mapping a control interface and managing it through GBB for example.
  - Detection of changes of GPIO input status signals could also be done by memory mapping and GBB, by constant polling.
- However, such implementation would not be standard, every vendor having its own. It also consumes extra resources:
  - This approach would require polling FSM on the host.
  - It would consume high bandwidth if latency needs to be low. Considering polling packet length of just 64 bytes and 10us polling around 50Mbps bandwidth per direction would be needed.

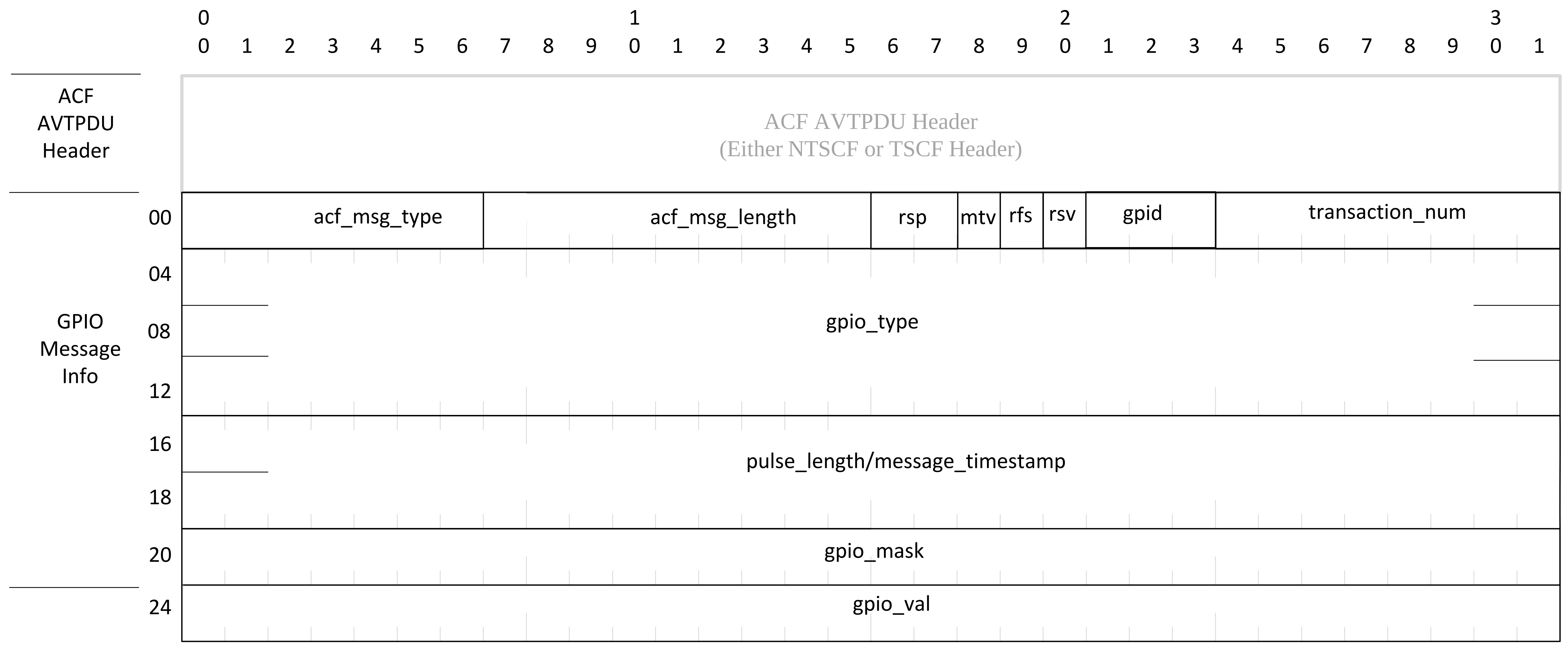
GPIO ACF packet format characteristics I

- *A new ACF format is proposed dedicated to GPIO encapsulation.*
- *The GPIO encapsulation is not designed to transfer high frequency GPIO changes or updates, but relatively low frequency GPIO updates (either inputs or outputs), potentially having tight timing/latency requirements.*
- *Can support, if needed, of several GPIO groups, identified by gpid field.*
- *Each GPIO group comprises 32 GPIOs. Each ACF message provides values for the 32 GPIOs within a group.*
- *Thanks to TSCF it is possible to synchronize changes on GPIO outputs using the presentation timestamp on the TSCF header. It is also possible to use NTSCF header when synchronization is not needed.*

GPIO ACF packet format characteristics II

- *Supports timestamping to indicate the timestamp where the GPIO inputs from the sensor were sampled.*
- *Possible to generate pulses on GPIO outputs with single message.*
- *Robust against packet loss and/or drop packets for latched GPIO inputs.*
- *Defines two type of responses:*
  - *Synchronous responses: sent right after execution of the action indicated on the request message.*
  - *Asynchronous responses: linked to a detection of a change on a GPIO input.*
    - *Frequency of generation of asynchronous responses would not be part of ACF specification, it is handled by “system integrator”.*

GPIO ACF proposal


0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
ACF AVTPDU Header
ACF AVTPDU Header (Either NTSCF or TSCF Header)
GPIO Message Info
00 acf_msg_type acf_msg_length rsp mtv rfs rsv gpid transaction_num
04
08 gpio_type
12
16 pulse_length/message_timestamp
18
20 gpio_mask
24 gpio_val

GPIO ACF packet fields I

- *acf_msg_type → ACF type.*
- *acf_msg_length → message length in quadlets. Fixed to 7.*
- *rsp → response, 2 bits. Differentiates between the three message types:*
  - *00_2 → Request messages from host to the sensor.*
  - *01_2 → Response. Messages sent by sensor after receiving a request message from the host.*
  - *10_2 → Refresh response. Messages sent by sensor to update the status of one of the GPIOs that are configured as inputs due to a change in its status.*
  - *11_2 → Reserved.*
- *mtv → message timestamp valid. Set to 0 on request messages. Set to 0 or 1 on response/refresh response. When set to 1 in a response message the pulse_length/message_timestamp field carries the acquisition/sampling time.*
- *rfs → refresh enable. When set to 1 on request messages, it enables the sending of refresh response messages. When set to 0, it disables the sending of refresh response messages. The interval between response/refresh responses is configured by the system integrator, so it is outside the scope of the 1722 standard.*

GPIO ACF packet fields II

- *rsv → 1 bit. Reserved.*
- *gpid → 3 bits. GPIO group ID. Up-to 8 groups of 32 GPIOs can be managed.*
- *transaction_num → identifies the message in the GPIO message stream. On response and refresh response messages the transaction_num of the last correctly received request message is used.*
- *gpio_type → 96 bits in total. 3 bits per GPIO. The direction of each GPIO is set according to the 1722 agent on the sensor node, so GPIO outputs are controlled by the talker (Host), and GPIO input value changes are communicated by the sensor. For each GPIO the meaning of the 3 bits is as follows:*
  - *000_2 → Normal input.*
  - *001_2 → Latched low input.*
  - *010_2 → Latched high input.*
  - *011_2 → Normal output.*
  - *100_2 → Pulsed output.*
  - *other → Reserved.*
- *gpio_mask → one bit per GPIO. Only used for outputs. If 0 it indicates that the corresponding GPIO is not addressed by this message. If 1, it indicates that the GPIO output value must be updated according to the value indicated by gpio_val.*

GPIO ACF packet fields III

- *pulse_length/timestamp→ in request messages, for pulsed output GPIOs this field indicates the period of the pulse to be generated. The value of the pulse is indicated in ns.*
- *gpio_val → one bit per GPIO. Meaning of the bit is different depending on the type of message (request or response/refresh response) and the type of GPIO.*
  - *On request messages, for output or pulsed output GPIOs, this field indicates the value to set the GPIO on the sensor when the message is received by the sensor (or at the presentation time when the message is carried on a TSCF header).*
    - *When the GPIO type is pulsed GPIO, the value on the GPIO should transition to its inverted value once the configured pulse-length have elapsed.*
  - *On request messages, for latched high or latched low GPIOs, gpio_val indicates the last value for those inputs that have been received by the Host.*
    - *The sensor can use this information as acknowledge of the input value transmission, so it can start sampling again the input value.*
  - *On response messages, for normal input GPIOs the sensor indicates the last sampled value for the corresponding GPIO.*
  - *On response messages, for latched input GPIOs, the sensor also indicates the GPIO value, but a latched functionality is implemented as indicated on the next slides.*

Latched functionality

- *As indicated previously the proposed GPIO protocol includes special GPIO type that ensures that critical state of the GPIO is transferred from sensor to the host even in presence of packet loss.*
- *When a GPIO is defined as latched low and the GPIO is sampled with low logic value on the sensor, the sensor will start reporting 0 value on the corresponding bit of gpio_val field on every response and refresh response message, until after a request message is received from the host having the gpio_val value of 0 on that bit. When such request message is received, the sensor can start sending the value according to the current status of the input.*
- *When a GPIO is defined as latched high the behavior is the same but being the critical state a logic high input value.*

Knowledge Development

Thank you!