January 2021

Standard Radar API: Proposal Version 0.1

Andrew Felch
Google LLC

Ivan Poupyrev
Google LLC

Shang Shi
Google LLC

Zhuo Wang
Google LLC

Chris Findeisen
Google LLC

See next page for additional authors

Inventor(s)
Andrew Felch, Ivan Poupyrev, Shang Shi, Zhuo Wang, Chris Findeisen, JinJie Chen, Blake Charles Jacquot, Mingyu Kang, Chang Hong Ye, Jaime Lien, Gerard George Pallipuram, Alex Paniutin, and Brandon Charles Barbello

Standard Radar API: Proposal Version 0.1

Abstract:

This publication describes a radar API that enables different radar options to be used without changing higher level software layers. The radar API supports frequency-modulated continuous-wave (FMCW) radar and other radar types. The radar API also supports interleaved radar configurations, so that different features may use different burst configurations simultaneously. For example, the radar API may support a dual burst configuration, where a first configuration is optimized for the longest range and a second configuration uses a wider bandwidth optimized for better range resolution. Together, the burst configurations may run simultaneously so that the algorithms supporting different features may consume their optimal input stream.

Keywords:

API, radar sensor, configuration, data, latency, frequency-modulated continuous-wave (FMCW) radar, interleaved radar configurations, burst configuration, range, resolvability

Background:

Future products will need the flexibility to choose amongst multiple radar options without placing an undue burden on software platform development. Configuring a radar sensor with a desired set of parameters and reading its data at low latency are the primary functions. The sensor may be interfaced with a digital signal processor (DSP) (on-chip or off-chip) with low-latency response to interrupt request (IRQ) (enabling smaller on-sensor buffers), or it may be run in a Linux driver with slower IRQ response rate (requiring larger buffers). An Application

Programming Interface (API) that enables different radar options to be used without changing higher level software layers is desired for this purpose.

Description:

This publication describes a proposed standard radar API. The radar API in this publication primarily focuses on frequency-modulated continuous-wave (FMCW) radar; support for other radar types is in-process. A desirable feature of the radar API is to support interleaved radar configurations, so that different features may use different burst configurations simultaneously. For example, one burst configuration may be optimized for the longest range, while a second configuration might use a wider bandwidth optimized for better range resolution. Together they may run simultaneously so that the algorithms supporting different features may consume their optimal input stream. This gives range benefits to a long-range feature, while giving resolvability benefits another feature. For power savings, one of the configurations might be disabled for a period of time. The configurations may be enabled/disabled independently so that, for example, the radar can run just a first burst type, just a second burst type, or simultaneously it might run both burst types. Transitions that enable or disable one configuration would not affect the continuous operation of the other enabled configurations. These use cases are just examples, and other radar configurations supporting other radar use-cases might be enabled or disabled on-the-fly as well.

Although the radar API in this early version is in C++, we anticipate it will be defined in an Interface Description Language (IDL) to enable any language to use the radar API. For simplicity we assume the interface library can directly perform the serial peripheral interface (SPI)

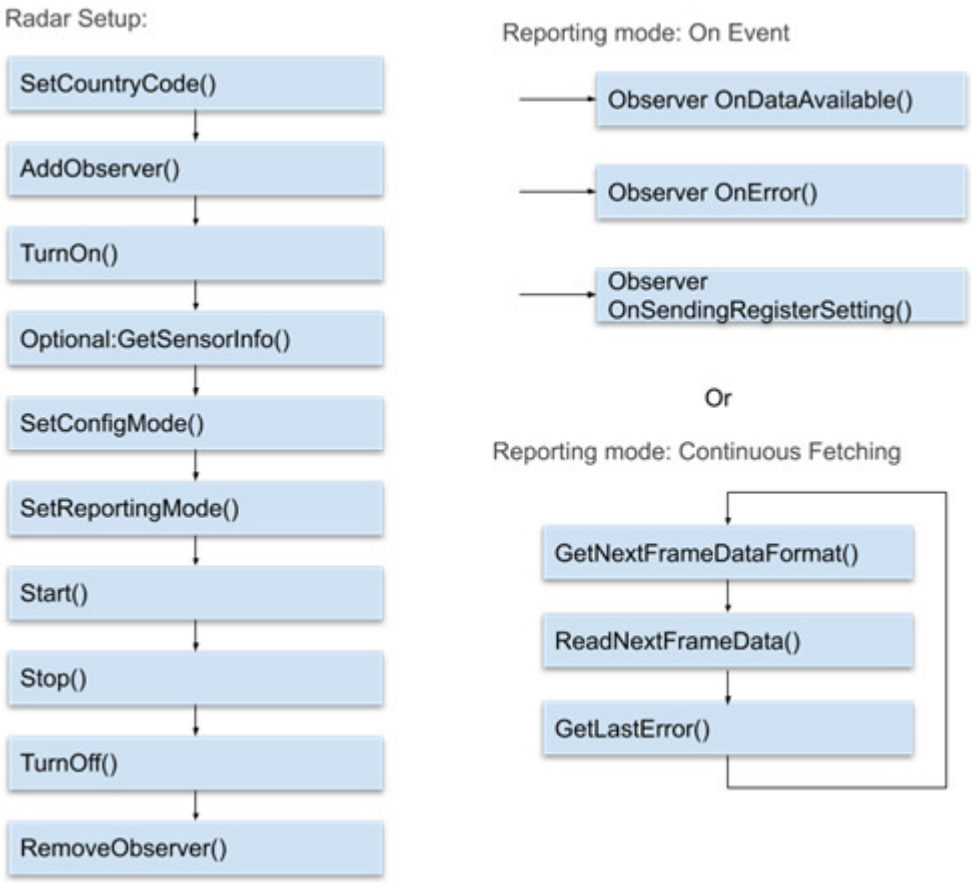
read/writes (or has other means of accessing data from the hardware) and receives interrupts from hardware when data is available, as necessary.

Example Usage:

```
radar_sensor_demo_main.cc:
// Radar API demo binary.

#include <string>
#include <vector>
#include "radar_sensor.h"
#include "radar_sensor_demo.h"
#include "sleep.h"

namespace {
using ::radar_sensor::ConfigMode;
using ::radar_sensor::ReportingMode;
}  // namespace

int main(int argc, char* argv[]) {

 // Create demo sensor.
 radar_sensor::RadarSensorDemo radar_sensor_demo;
 radar_sensor::RadarSensor* sensor = &radar_sensor_demo;

 // Create demo sensor observer.
 radar_sensor::RadarSensorObserverDemo
radar_sensor_observer_demo;
 radar_sensor::RadarSensorObserver* observer =
&radar_sensor_observer_demo;

 // Set country code.
 std::string country_code = "US";
 sensor->SetCountryCode(country_code);

 // Add observer to sensor.
 sensor->AddObserver(observer);

 // Turn on radar sensor.
 sensor->TurnOn();
```

```
 // Set sensor to predefined mode.
 sensor->SetConfigMode(std::vector<ConfigMode>{
     ConfigMode::kLongRangeMode,
ConfigMode::kHighRangeResolutionMode});

 // Set sensor reporting mode. Here uses on-event to link with
observer.
 sensor->SetReportingMode(ReportingMode::kOnEvent);

 // Start sensor to stream data.
 sensor->Start();

 SleepForMilliseconds(1000);

 // Stop sensor and go to idle.
 sensor->Stop();

 // Turn off sensor.
 sensor->TurnOff();

 // Remove observer from sensor.
 sensor->RemoveObserver(observer);

 return 0;
}
```

Radar API:



Radar API Extended:

radar_sensor::RadarSensor

Radar sensor controller class. Target user: user applications.

Public functions	
AddObserver(RadarSensorObserver *observer)=0	virtual Status Adds data/event observers.
GetLastError()=0	virtual Status Gets chip last error.
GetNextFrameDataFormat()=0	virtual DataFrameFormat Gets next frame data format.
GetSensorInfo()=0	virtual Info Gets chip info.
ReadNextFrameData(std::unique_ptr< std::vector< uint8_t >> raw_radar_data)=0	virtual Status Read data if available.
RemoveObserver(RadarSensorObserver *observer)=0	virtual Status Removes event observers.
SetConfigMode(std::vector< ConfigMode > enabled_config_modes)=0	virtual Status Enables multiple config modes.
SetCountryCode(const std::string & country_code)=0	virtual Status Check if current location's ISO 3166-1 alpha-2 country code is allowed.
SetReportingMode(ReportingMode reporting_mode)=0	virtual Status Sets event reporting mode.
Start()=0	virtual Status Starts radar chip TX, RX and data/error streaming.

Stop()=0	virtual Status Stops radar chip TX, RX and data/error streaming. May join threads for data/error streaming.
TurnOff()=0	virtual Status Cleans up and power off radar sensor. Needs to be blocking.
TurnOn()=0	virtual Status Powers on sensor chip and/or setup memory and threads if needed.

Public functions

AddObserver

virtual Status AddObserver(RadarSensorObserver *observer)=0

Adds data/event observers. Only used in on-event reporting mode. Observers need to be added before sensor is STARTED. Observer needs to be alive until observer is removed.

GetLastError

virtual Status GetLastError()=0 virtual Status AddObserver(RadarSensorObserver *observer)=0

Gets chip last error.

GetNextFrameDataFormat

virtual DataFrameFormat GetNextFrameDataFormat()=0

Gets next frame data format.

Only available in continuous-fetching reporting mode. Radar sensor must be ON and STARTED.

GetSensorInfo

virtual Info GetSensorInfo()=0

Gets chip info.

ReadNextFrameData

virtual Status ReadNextFrameData(std::unique_ptr< std::vector< uint8_t >> raw_radar_data)=0

Reads data if available.

Radar sensor must be ON and STARTED before data reading. Only available in continuous-fetching reporting mode.

RemoveObserver

virtual Status RemoveObserver(RadarSensorObserver *observer)=0

Removes event observers.

Observers need to be removed after sensor is STOPPED.

SetConfigMode

virtual Status SetConfigMode(std::vector< ConfigMode > enabled_config_modes)=0

Enables multiple config modes.

Radar must be ON and STOPPED to be configured.

SetCountryCode

virtual Status SetCountryCode(const std::string & country_code)=0

Check if current location's ISO 3166-1 alpha-2 country code is allowed.

If not, radar will not turn on.

SetReportingMode

virtual Status SetReportingMode(ReportingMode reporting_mode)=0

Sets event reporting mode.

Observers will only be used in on-event reporting mode.

Start

virtual Status Start()=0

Starts radar chip TX, RX and data/error streaming.

May create threads for data/error streaming.

Stop

virtual Status Stop()=0

Stops radar chip TX, RX and data/error streaming. May join threads for data/error streaming.

TurnOff

virtual Status TurnOff()=0

Cleans up and power off radar sensor. Needs to be blocking.

TurnOn

virtual Status TurnOn()=0

Powers on sensor chip and/or setup memory and threads if needed.

If country code is not provided or invalid, radar should not turn on. Turn on needs to be synchronous/blocking.

radar_sensor::RadarSensorExtended

Radar sensor controller class with extended options. Target user: Initial development / testing / verification.

Public functions	
AddRadarConfig(RadarConfig config)=0	virtual uint32_t Saves radar config for later use. Returns config index.
DisableAllRadarConfigs()=0	virtual Status Disables all radar configs.
EnableRadarConfig(std::vector< uint32_t > config_indexes)=0	virtual Status Enables radar configs with selected index.
GetRadarConfigIndexes()=0	virtual std::vector< uint32_t > Get saved radar config indexes.
ReadRadarConfig(uint32_t config_index)=0	virtual RadarConfig Reads radar config details.
ReadRegisters()=0	virtual std::vector< uint32_t > Reads from radar chip registers.
RemoveRadarConfig(uint32_t config_index)=0	virtual Status Removes saved radar config by index.
SetRegisters(std::vector< RegisterSetting > setting)=0	virtual Status Writes to radar chip registers.

Public functions

AddRadarConfig

virtual uint32_t AddRadarConfig(RadarConfig config)=0

Saves radar config for later use. Returns config index.

DisableAllRadarConfigs

virtual Status DisableAllRadarConfigs()=0

Disables all radar configs.

EnableRadarConfig

virtual Status EnableRadarConfig(std::vector< uint32_t > config_indexes)=0

Enables radar configs with selected index.

GetRadarConfigIndexes

virtual std::vector< uint32_t > GetRadarConfigIndexes()=0

Get saved radar config indexes.

ReadRadarConfig

virtual RadarConfig ReadRadarConfig(uint32_t config_index)=0

Reads radar config details.

ReadRegisters

virtual std::vector< uint32_t > ReadRegisters()=0

Reads from radar chip registers.

Only available when radar is ON and STOPPED.

RemoveRadarConfig

virtual Status RemoveRadarConfig(uint32_t config_index)=0

Removes saved radar config by index.

SetRegisters

virtual Status SetRegisters(std::vector< RegisterSetting > setting)=0

Writes to radar chip registers.

Only available when radar is ON and STOPPED.

radar_sensor::RadarSensorObserver

Radar sensor data/event observer class. Observer interface for on-event reporting mode. Callbacks are only called when radar sensor is ON and STARTED. Callbacks should be handled asynchronously.

Public functions	
OnDataAvailable(std::unique_ptr< std::vector< uint8_t >> raw_radar_data, DataFrameFormat format)=0	virtual void Called when there is new available data. Handles data asynchronously.
OnError(Status status)=0	virtual void Handles latest available error asynchronously.
OnSendingRegisterSetting(RegisterSetting)=0	virtual void This callback is required to be called when register setting is sent to radar device.

Public functions

OnDataAvailable

virtual void OnDataAvailable(std::unique_ptr< std::vector< uint8_t >> raw_radar_data, DataFrameFormat format)=0

Called when there is new available data. Handles data asynchronously.

OnError

virtual void OnError(Status status)=0

Handles latest available error asynchronously.

OnSendingRegisterSetting

virtual void OnSendingRegisterSetting(RegisterSetting)=0

This callback is required to be called when register setting is sent to radar device.

API Demo

Example output:

```
I1005 09:23:53.266068 3375274 init-domain.cc:39] Fiber init: default domain = pthread, concurrency = 79

I1005 09:23:53.266364 3375274 radar_sensor_demo.cc:73] [Radar Sensor]: Country code set to:"US".

I1005 09:23:53.266492 3375274 radar_sensor_demo.cc:86] [Radar Sensor]: Observer added.

I1005 09:23:53.266522 3375274 radar_sensor_demo.cc:119] [Radar Sensor]: Radar turned on.

I1005 09:23:53.266558 3375274 radar_sensor_demo.cc:142] [Radar Sensor]: Radar config mode set to: "1,3".

I1005 09:23:53.266592 3375274 radar_sensor_demo.cc:153] [Radar Sensor]: Radar reporting mode set to: "on_event".

I1005 09:23:53.266770 3375274 radar_sensor_demo.cc:180] [Radar Sensor]: Radar data streaming started.

I1005 09:23:53.267721 3375282 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:high_range_resolution_mode [0]:9 [1]:3 ... [end]: 4

I1005 09:23:53.267978 3375283 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:long_range_mode [0]:9 [1]:9 ... [end]: 6

I1005 09:23:53.301075 3375282 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:high_range_resolution_mode [0]:4 [1]:8 ... [end]: 2

I1005 09:23:53.334406 3375282 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:high_range_resolution_mode [0]:7 [1]:6 ... [end]: 1

I1005 09:23:53.367764 3375282 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:high_range_resolution_mode [0]:8 [1]:2 ... [end]: 5

I1005 09:23:53.368305 3375283 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:long_range_mode [0]:7 [1]:10 ... [end]: 7

I1005 09:23:53.401138 3375282 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:high_range_resolution_mode [0]:4 [1]:3 ... [end]: 9

I1005 09:23:53.434469 3375282 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:high_range_resolution_mode [0]:7 [1]:5 ... [end]: 8

I1005 09:23:53.467863 3375282 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:high_range_resolution_mode [0]:3 [1]:6 ... [end]: 7

I1005 09:23:53.468580 3375283 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:long_range_mode [0]:8 [1]:6 ... [end]: 2

I1005 09:23:53.501240 3375282 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:high_range_resolution_mode [0]:4 [1]:9 ... [end]: 1
```

```
I1005 09:23:53.534625 3375282 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:high_range_resolution_mode [0]:4 [1]:7 ... [end]: 9
I1005 09:23:53.568005 3375282 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:high_range_resolution_mode [0]:2 [1]:9 ... [end]: 5
I1005 09:23:53.568967 3375283 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:long_range_mode [0]:9 [1]:3 ... [end]: 2
I1005 09:23:53.601386 3375282 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:high_range_resolution_mode [0]:5 [1]:10 ... [end]: 8
I1005 09:23:53.634736 3375282 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:high_range_resolution_mode [0]:6 [1]:10 ... [end]: 6
I1005 09:23:53.668943 3375282 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:high_range_resolution_mode [0]:3 [1]:1 ... [end]: 5
I1005 09:23:53.669367 3375283 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:long_range_mode [0]:6 [1]:5 ... [end]: 7
I1005 09:23:53.702233 3375282 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:high_range_resolution_mode [0]:3 [1]:3 ... [end]: 1
I1005 09:23:53.735571 3375282 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:high_range_resolution_mode [0]:2 [1]:8 ... [end]: 8
I1005 09:23:53.768917 3375282 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:high_range_resolution_mode [0]:8 [1]:6 ... [end]: 10
I1005 09:23:53.769704 3375283 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:long_range_mode [0]:5 [1]:4 ... [end]: 2
I1005 09:23:53.802270 3375282 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:high_range_resolution_mode [0]:8 [1]:10 ... [end]: 4
I1005 09:23:53.835601 3375282 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:high_range_resolution_mode [0]:5 [1]:6 ... [end]: 9
I1005 09:23:53.868893 3375282 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:high_range_resolution_mode [0]:9 [1]:2 ... [end]: 6
I1005 09:23:53.870045 3375283 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:long_range_mode [0]:6 [1]:9 ... [end]: 5
I1005 09:23:53.902228 3375282 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:high_range_resolution_mode [0]:3 [1]:7 ... [end]: 2
I1005 09:23:53.935596 3375282 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:high_range_resolution_mode [0]:4 [1]:4 ... [end]: 5
I1005 09:23:53.968969 3375282 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:high_range_resolution_mode [0]:5 [1]:2 ... [end]: 7
I1005 09:23:53.970412 3375283 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:long_range_mode [0]:7 [1]:2 ... [end]: 3
```

```
I1005 09:23:54.002304 3375282 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:high_range_resolution_mode [0]:6 [1]:10 ... [end]: 4

I1005 09:23:54.035658 3375282 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:high_range_resolution_mode [0]:10 [1]:9 ... [end]: 1

I1005 09:23:54.069001 3375282 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:high_range_resolution_mode [0]:3 [1]:7 ... [end]: 10

I1005 09:23:54.070742 3375283 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:long_range_mode [0]:9 [1]:3 ... [end]: 4

I1005 09:23:54.102358 3375282 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:high_range_resolution_mode [0]:10 [1]:2 ... [end]: 2

I1005 09:23:54.135697 3375282 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:high_range_resolution_mode [0]:8 [1]:9 ... [end]: 5

I1005 09:23:54.168989 3375282 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:high_range_resolution_mode [0]:1 [1]:6 ... [end]: 1

I1005 09:23:54.171075 3375283 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:long_range_mode [0]:10 [1]:3 ... [end]: 2

I1005 09:23:54.202329 3375282 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:high_range_resolution_mode_mode [0]:7 [1]:5 ... [end]: 4

I1005 09:23:54.235609 3375282 radar_sensor_demo.cc:56] [Observer]: Data received: [mode]:high_range_resolution_mode_mode [0]:4 [1]:8 ... [end]: 4

I1005 09:23:54.271555 3375274 radar_sensor_demo.cc:191] [Radar Sensor]: Radar data streaming stopped.

I1005 09:23:54.271685 3375274 radar_sensor_demo.cc:134] [Radar Sensor]: Radar turned off.

I1005 09:23:54.271785 3375274 radar_sensor_demo.cc:99] [Radar Sensor]: Observer removed.
```