

v4l2

Claude Heiland-Allen

2011–2019

# Contents

|    |                                                   |    |
|----|---------------------------------------------------|----|
| 1  | Foreign/Extra/BitSet.hs . . . . .                 | 2  |
| 2  | Foreign/Extra/CEnum.hs . . . . .                  | 3  |
| 3  | Foreign/Extra/String.hs . . . . .                 | 3  |
| 4  | .gitignore . . . . .                              | 3  |
| 5  | Graphics/V4L2/Capability.hs . . . . .             | 3  |
| 6  | Graphics/V4L2/ColorSpace.hs . . . . .             | 6  |
| 7  | Graphics/V4L2/ColorSpace/Internal.hs . . . . .    | 6  |
| 8  | Graphics/V4L2/Control.hs . . . . .                | 7  |
| 9  | Graphics/V4L2/Device.hs . . . . .                 | 10 |
| 10 | Graphics/V4L2/Field.hs . . . . .                  | 11 |
| 11 | Graphics/V4L2/Field/Internal.hs . . . . .         | 12 |
| 12 | Graphics/V4L2/Format.hs . . . . .                 | 13 |
| 13 | Graphics/V4L2.hs . . . . .                        | 19 |
| 14 | Graphics/V4L2/IOCtl.hs . . . . .                  | 21 |
| 15 | Graphics/V4L2/PixelFormat.hs . . . . .            | 22 |
| 16 | Graphics/V4L2/PixelFormat/Internal.hs . . . . .   | 22 |
| 17 | Graphics/V4L2/Priority.hs . . . . .               | 25 |
| 18 | Graphics/V4L2/Types.hs . . . . .                  | 26 |
| 19 | Graphics/V4L2/Types/Internal.hs . . . . .         | 26 |
| 20 | Graphics/V4L2/VideoCapture.hs . . . . .           | 27 |
| 21 | Graphics/V4L2/VideoCapture/Read.hs . . . . .      | 27 |
| 22 | Graphics/V4L2/VideoInput.hs . . . . .             | 28 |
| 23 | Graphics/V4L2/VideoStandard.hs . . . . .          | 31 |
| 24 | Graphics/V4L2/VideoStandard/Internal.hs . . . . . | 31 |
| 25 | LICENSE . . . . .                                 | 37 |
| 26 | Setup.hs . . . . .                                | 37 |
| 27 | TODO . . . . .                                    | 37 |
| 28 | v4l2.cabal . . . . .                              | 39 |

## 1 Foreign/Extra/BitSet.hs

```

module Foreign.Extra.BitSet
  ( fromBitSet
  , toBitSet
  ) where

5
import Data.Bits (Bits, (.&.), (.|.), complement)
import Data.Set (Set, empty, insert, toList)

fromBitSet :: (Ord h, Bits c, Num c) => [(h, c)] -> (c -> h) -> c -> Set h
10 fromBitSet spec unknown bits

```

```

    | unmatched == 0 = matched
    | otherwise = insert (unknown unmatched) matched
  where
    (matched, unmatched) = foldr test (empty, bits) spec
15   test (h, v) (m, um)
        | v == v .&. um = (insert h m, um .&. complement v)
        | otherwise    = (m, um)

toBitSet :: (Eq h, Bits c, Num c) => [(h, c)] -> (h -> Bool) -> (h -> c) -> Set h
    ↪ -> c
20 toBitSet spec isUnknown unUnknown = foldr (.|.) 0 . map f . toList
    where
      f h
        | isUnknown h = unUnknown h
        | otherwise = case h 'lookup' spec of
25           Just c -> c
           Nothing -> error "toBitSet"

```

## 2 Foreign/Extra/CEnum.hs

```

module Foreign.Extra.CEnum
  ( fromCEnum
    , toCEnum
    ) where
5
fromCEnum :: (Ord h, Eq c) => [(h, c)] -> (c -> h) -> c -> h
fromCEnum spec unknown enum = case enum 'lookup' map swap spec of
    Nothing -> unknown enum
    Just h -> h
10 where
    swap (a, b) = (b, a)

toCEnum :: Eq h => [(h, c)] -> (h -> Bool) -> (h -> c) -> h -> c
toCEnum spec isUnknown unUnknown h
15   | isUnknown h = unUnknown h
   | otherwise = case h 'lookup' spec of
       Nothing -> error "toCEnum"
       Just c -> c

```

## 3 Foreign/Extra/String.hs

```

module Foreign.Extra.String
  ( fromString
    ) where
5
import Data.Word (Word8)

fromString :: [Word8] -> String
fromString = map (toEnum . fromEnum) . takeWhile (/= 0)

```

## 4 .gitignore

```
dist
```

## 5 Graphics/V4L2/Capability.hs

```

{-# LANGUAGE DeriveDataTypeable #-}
{- |
Module      : Graphics.V4L2.Capability
Maintainer  : claud@mathr.co.uk
5  Stability : no
Portability : no
-}
module Graphics.V4L2.Capability
  ( DeviceInfo(..)
10   , DeviceCapability(..)
    , deviceInfoCapabilities
    , deviceInfo
    , deviceCapabilities
  ) where

15  import Data.Bits ((.&.), shiftR)
  import Data.Data (Data)
  import Data.Set (Set)
  import Data.Typeable (Typeable)
20  import Data.Version (Version(..))
  import Data.Word (Word32)

  import Bindings.Linux.VideoDev2

25  import Foreign.Extra.BitSet (fromBitSet)
  import Foreign.Extra.String (fromString)
  import Graphics.V4L2.Device (Device)
  import Graphics.V4L2.IOCtl (ioctl ')

30  {- | Device information. -}
  data DeviceInfo = DeviceInfo{ deviceDriver , deviceCard , deviceBus :: String , ↵
    ↵ deviceVersion :: Version }
    deriving (Eq, Ord, Read, Show, Typeable)

  {- | Device capabilities. -}
35  data DeviceCapability
    = DeviceVideoCapture
    | DeviceVideoOutput
    | DeviceVideoOverlay
    | DeviceVbiCapture
40  | DeviceVbiOutput
    | DeviceSlicedVbiCapture
    | DeviceSlicedVbiOutput
    | DeviceRdsCapture
    | DeviceVideoOutputOverlay
45  | DeviceHwFreqSeek
    | DeviceRdsOutput
    | DeviceTuner
    | DeviceAudio
    | DeviceRadio
50  | DeviceModulator
    | DeviceReadWrite
    | DeviceAsyncIO
    | DeviceStreaming
    | DeviceTimePerFrame
55  | DeviceUnknown Word32
    deriving (Eq, Ord, Read, Show, Data, Typeable)

```

```

fromCapability :: Word32 -> Set DeviceCapability
fromCapability = fromBitSet
60   [ ( DeviceVideoCapture      , c'V4L2_CAP_VIDEO_CAPTURE      )
      , ( DeviceVideoOutput     , c'V4L2_CAP_VIDEO_OUTPUT     )
      , ( DeviceVideoOverlay    , c'V4L2_CAP_VIDEO_OVERLAY    )
      , ( DeviceVbiCapture      , c'V4L2_CAP_VBI_CAPTURE      )
      , ( DeviceVbiOutput       , c'V4L2_CAP_VBI_OUTPUT       )
65   , ( DeviceSlicedVbiCapture  , c'V4L2_CAP_SLICED_VBI_CAPTURE  )
      , ( DeviceSlicedVbiOutput , c'V4L2_CAP_SLICED_VBI_OUTPUT )
      , ( DeviceRdsCapture      , c'V4L2_CAP_RDS_CAPTURE      )
      , ( DeviceVideoOutputOverlay , c'V4L2_CAP_VIDEO_OUTPUT_OVERLAY )
      , ( DeviceHwFreqSeek     , c'V4L2_CAP_HW_FREQ_SEEK     )
70   , ( DeviceRdsOutput        , c'V4L2_CAP_RDS_OUTPUT        )
      , ( DeviceTuner          , c'V4L2_CAP_TUNER          )
      , ( DeviceAudio          , c'V4L2_CAP_AUDIO          )
      , ( DeviceRadio          , c'V4L2_CAP_RADIO          )
      , ( DeviceModulator      , c'V4L2_CAP_MODULATOR      )
75   , ( DeviceReadWrite       , c'V4L2_CAP_READWRITE       )
      , ( DeviceAsyncIO        , c'V4L2_CAP_ASYNCIO        )
      , ( DeviceStreaming      , c'V4L2_CAP_STREAMING      )
      , ( DeviceTimePerFrame   , c'V4L2_CAP_TIMEPERFRAME   )
      ] DeviceUnknown
80
{- | Get device information and capabilities. -}
deviceInfoCapabilities :: Device -> IO (DeviceInfo, Set DeviceCapability)
deviceInfoCapabilities d = do
  c <- ioctl' d C'VIDIOC_QUERYCAP
85   return
      ( DeviceInfo
        { deviceDriver = fromString $ c'v4l2_capability'driver c
        , deviceCard = fromString $ c'v4l2_capability'card c
        , deviceBus = fromString $ c'v4l2_capability'bus_info c
90        , deviceVersion = fromVersion $ c'v4l2_capability'version c
        }
        , fromCapability $ c'v4l2_capability'capabilities c
        )

95   {- | Get device information. -}
deviceInfo :: Device -> IO DeviceInfo
deviceInfo d = fst 'fmap' deviceInfoCapabilities d

{- | Get device capabilities. -}
100  deviceCapabilities :: Device -> IO (Set DeviceCapability)
deviceCapabilities d = snd 'fmap' deviceInfoCapabilities d

fromVersion :: Word32 -> Version
fromVersion v = Version
105   { versionBranch =
      [ fromIntegral $ (v `shiftR` 16) .&. 0xFF
      , fromIntegral $ (v `shiftR` 8) .&. 0xFF
      , fromIntegral $ (v
110      , versionTags = []
      }

```

## 6 Graphics/V4L2/ColorSpace.hs

```

{- |
Module      : Graphics.V4L2.ColorSpace
Maintainer  : claudem@mathr.co.uk
Stability   : no
5  Portability : no
-}
module Graphics.V4L2.ColorSpace
  ( module Graphics.V4L2.ColorSpace.Internal
  ) where
10
import Graphics.V4L2.ColorSpace.Internal hiding (fromColorSpace, toColorSpace)

```

## 7 Graphics/V4L2/ColorSpace/Internal.hs

```

{-# LANGUAGE DeriveDataTypeable #-}
{- |
Module      : Graphics.V4L2.ColorSpace.Internal
Maintainer  : claudem@mathr.co.uk
5  Stability   : no
  Portability : no
-}
module Graphics.V4L2.ColorSpace.Internal
  ( ColorSpace(..)
10    , fromColorSpace
    , toColorSpace
    ) where

import Data.Data (Data)
15 import Data.Typeable (Typeable)
import Data.Word (Word32)

import Bindings.Linux.VideoDev2

20 import Foreign.Extra.CEnum (toCEnum, fromCEnum)

{- | Color spaces. -}
data ColorSpace
  = ColorSMPTE170M
25    | ColorSMPTE240M
    | ColorREC709
    | ColorBT878
    | Color470SystemM
    | Color470SystemBG
30    | ColorJPEG
    | ColorSRGB
    | ColorUnknown Word32
  deriving (Eq, Ord, Read, Show, Data, Typeable)

35 {- | internal -}
fromColorSpace :: C'v4l2_colorspace -> ColorSpace
{- | internal -}
toColorSpace :: ColorSpace -> C'v4l2_colorspace

40 (fromColorSpace, toColorSpace) = (fromCEnum spec (ColorUnknown . fromIntegral), \
  ↪ toCEnum spec isUnknown unUnknown) where

```

```

spec =
  [ ( ColorSMPTE170M    , c 'V4L2_COLORSPACE_SMPTE170M    )
    , ( ColorSMPTE240M    , c 'V4L2_COLORSPACE_SMPTE240M    )
    , ( ColorREC709       , c 'V4L2_COLORSPACE_REC709       )
45   , ( ColorBT878       , c 'V4L2_COLORSPACE_BT878       )
    , ( Color470SystemM   , c 'V4L2_COLORSPACE_470_SYSTEM_M   )
    , ( Color470SystemBG  , c 'V4L2_COLORSPACE_470_SYSTEM_BG  )
    , ( ColorJPEG        , c 'V4L2_COLORSPACE_JPEG        )
    , ( ColorSRGB        , c 'V4L2_COLORSPACE_SRGB        )
50   ]
isUnknown (ColorUnknown _) = True
isUnknown _ = False
unUnknown (ColorUnknown f) = fromIntegral f
unUnknown _ = error "Graphice.V4L2.ColorSpace.Internal.toColorSpace"

```

## 8 Graphics/V4L2/Control.hs

```

{-# LANGUAGE DeriveDataTypeable, GeneralizedNewtypeDeriving #-}
{- |
Module      : Graphics.V4L2.Control
Maintainer  : claude@mathr.co.uk
5  Stability : no
Portability : no
-}
module Graphics.V4L2.Control
  ( ControlID()
10   , MenuID()
    , ControlFlag(..)
    , ControlInfo(..)
    , queryControls
    , Activate(..)
15   , ControlData()
    , getControl
    , setControl
  ) where

20  import Control.Exception as E (catch, throwIO)
  import Control.Monad (liftM)
  import Data.Data (Data)
  import Data.Int (Int32)
  import Data.Map (Map)
25  import qualified Data.Map as M
  import Data.Maybe (catMaybes)
  import Data.Set (Set)
  import Data.Typeable (Typeable)
  import Data.Word (Word32)
30  import Foreign.Marshal.Utils (fromBool, toBool)
  import GHC.IO.Exception (IOErrorType(InvalidArgument), ioe_type)

  import Bindings.Linux.VideoDev2

35  import Foreign.Extra.BitSet (fromBitSet)
  import Foreign.Extra.String (fromString)
  import Graphics.V4L2.Device (Device)
  import Graphics.V4L2.IOctl (ioctl, ioctl_, zero)

40  {- | Control index. -}

```

```

newtype ControlID = ControlID Int
    deriving (Eq, Ord, Enum, Bounded, Read, Show, Data, Typeable, Num, Integral, ↗
        ↘ Real)

{- | Menu index. -}
45 newtype MenuID = MenuID Int
    deriving (Eq, Ord, Enum, Bounded, Read, Show, Data, Typeable, Num, Integral, ↗
        ↘ Real)

{- | Control flags. -}
data ControlFlag
50   = ControlDisabled
    | ControlGrabbed
    | ControlReadOnly
    | ControlUpdate
    | ControlInactive
55   | ControlSlider
    | ControlWriteOnly
    | ControlUnknown Word32
    deriving (Eq, Ord, Read, Show, Data, Typeable)

60 fromControlFlag :: Word32 -> Set ControlFlag
fromControlFlag = fromBitSet
    [ ( ControlDisabled  , c'V4L2.CTRL_FLAG_DISABLED  )
      , ( ControlGrabbed  , c'V4L2.CTRL_FLAG_GRABBED   )
      , ( ControlReadOnly , c'V4L2.CTRL_FLAG_READ_ONLY )
65   , ( ControlUpdate    , c'V4L2.CTRL_FLAG_UPDATE    )
      , ( ControlInactive , c'V4L2.CTRL_FLAG_INACTIVE  )
      , ( ControlSlider   , c'V4L2.CTRL_FLAG_SLIDER    )
      , ( ControlWriteOnly, c'V4L2.CTRL_FLAG_WRITE_ONLY )
      ] ControlUnknown

70 {- | Control information. -}
data ControlInfo
    = ControlInteger { controlName :: String, controlFlags :: Set ControlFlag, ↗
        ↘ controlDefaultInt, controlMinimum, controlMaximum, controlStep :: Int }
    | ControlBoolean { controlName :: String, controlFlags :: Set ControlFlag, ↗
        ↘ controlDefaultBool :: Bool }
75   | ControlButton { controlName :: String, controlFlags :: Set ControlFlag }
    | ControlInteger64 { controlName :: String, controlFlags :: Set ControlFlag }
    | ControlString { controlName :: String, controlFlags :: Set ControlFlag, ↗
        ↘ controlLengthWithout0 :: Int }
    | ControlMenu { controlName :: String, controlFlags :: Set ControlFlag, ↗
        ↘ controlDefaultMenu :: MenuID, controlMenu :: Map MenuID String }
    deriving (Eq, Ord, Read, Show, Data, Typeable)

80 {- | Enumerate controls. -}
queryControls :: Device -> IO (Map ControlID ControlInfo)
queryControls d = do
    cs1 <- (M.fromList . catMaybes) 'fmap' mapM (queryctrl d) [c'V4L2_CID_BASE .. ↗
        ↘ c'V4L2_CID_LASTP1 - 1]
85   cs2 <- M.fromList 'fmap' unfoldM q (c'V4L2_CID_PRIVATE_BASE)
    return $ cs1 'M.union' cs2
    where
        q c = fmap (flip (,) (c + 1)) 'fmap' queryctrl d c

90 queryctrl :: Device -> ControlID -> IO (Maybe (ControlID, ControlInfo))

```

```

queryctrl d n = do
  mq <- (return . Just <<< ioctl d C'VIDIOC_QUERYCTRL . (\s->s{ c'v4l2_queryctrl ↵
    ↵ 'id = fromIntegral n }) <<< zero
    ) 'E.catch' (\e -> case ioe_type e of
      InvalidArgument -> return Nothing
95     _ -> throwIO e)
  case mq of
    Nothing -> return Nothing
    Just q -> return . Just . (,) n <<< case c'v4l2_queryctrl'type q of
      x | x == c'V4L2_CTRL_TYPE_INTEGER -> return ControlInteger
100        { controlName = fromString $ c'v4l2_queryctrl'name q
          , controlFlags = fromControlFlag $ c'v4l2_queryctrl'flags q
          , controlDefaultInt = fromIntegral $ c'v4l2_queryctrl'default_value ↵
            ↵ q
          , controlMinimum = fromIntegral $ c'v4l2_queryctrl'minimum q
          , controlMaximum = fromIntegral $ c'v4l2_queryctrl'maximum q
105        , controlStep = fromIntegral $ c'v4l2_queryctrl'step q }
      | x == c'V4L2_CTRL_TYPE_BOOLEAN -> return ControlBoolean
        { controlName = fromString $ c'v4l2_queryctrl'name q
          , controlFlags = fromControlFlag $ c'v4l2_queryctrl'flags q
          , controlDefaultBool = toBool $ c'v4l2_queryctrl'default_value q }
110      | x == c'V4L2_CTRL_TYPE_BUTTON -> return ControlButton
        { controlName = fromString $ c'v4l2_queryctrl'name q
          , controlFlags = fromControlFlag $ c'v4l2_queryctrl'flags q }
      | x == c'V4L2_CTRL_TYPE_INTEGER64 -> return ControlInteger64
        { controlName = fromString $ c'v4l2_queryctrl'name q
115        , controlFlags = fromControlFlag $ c'v4l2_queryctrl'flags q }
      | x == c'V4L2_CTRL_TYPE_STRING -> return ControlString
        { controlName = fromString $ c'v4l2_queryctrl'name q
          , controlFlags = fromControlFlag $ c'v4l2_queryctrl'flags q
          , controlLengthWithout0 = fromIntegral $ c'v4l2_queryctrl'maximum q ↵
            ↵ }
120      | x == c'V4L2_CTRL_TYPE_MENU -> (\m -> ControlMenu
        { controlName = fromString $ c'v4l2_queryctrl'name q
          , controlFlags = fromControlFlag $ c'v4l2_queryctrl'flags q
          , controlDefaultMenu = fromIntegral $ c'v4l2_queryctrl'default_value ↵
            ↵ q
          , controlMenu = m }) 'fmap' querymenus d n (fromIntegral $ c'↵
            ↵ v4l2_queryctrl'minimum q) (fromIntegral $ c'v4l2_queryctrl'↵
            ↵ maximum q)
125      | otherwise -> error "queryctrl"

-- FIXME: V4L2_CTRL_TYPE_CTRL_CLASS

130 querymenus :: Device -> ControlID -> MenuID -> MenuID -> IO (Map MenuID String)
querymenus d c mi ma = (M.fromList . catMaybes) 'fmap' mapM (querymenu d c) [mi ↵
  ↵ .. ma]

querymenu :: Device -> ControlID -> MenuID -> IO (Maybe (MenuID, String))
querymenu d c m = do
135   i <- (return . Just <<< ioctl d C'VIDIOC_QUERYMENU . (\s->s{ c'v4l2_querymenu'↵
    ↵ id = fromIntegral c, c'v4l2_querymenu'index = fromIntegral m }) <<< zero
    ) 'E.catch' (\e -> case ioe_type e of
      InvalidArgument -> return Nothing
      _ -> throwIO e)
  return $ case i of

```

```

140     Nothing -> Nothing
        Just fs -> Just (m, fromString $ c'v4l2_querymenu' name fs)

{- | Control data values -}
class ControlData d where
145   toControlData :: d -> Int32
        fromControlData :: Int32 -> d

instance ControlData Bool where
        toControlData = fromBool
150   fromControlData = toBool

instance ControlData MenuID where
        toControlData = fromIntegral
        fromControlData = fromIntegral
155
instance ControlData Int32 where
        toControlData = id
        fromControlData = id

160 {- | Button control data. -}
data Activate = Activate
        deriving (Eq, Ord, Enum, Bounded, Read, Show, Data, Typeable)

instance ControlData Activate where
165   toControlData Activate = 0
        fromControlData _ = Activate

{- | Get a control. -}
getControl :: ControlData d => Device -> ControlID -> IO d
170 getControl d c = return . fromControlData . c'v4l2_control'value ==<< ioctl d C'V
        \< VIDIOC_G_CTRL . (\s->s{ c'v4l2_control'id = fromIntegral c }) ==<< zero

{- | Set a control. -}
setControl :: ControlData d => Device -> ControlID -> d -> IO ()
setControl d c x = ioctl_ d C'VIDIOC_S_CTRL . (\s->s{ c'v4l2_control'id = <
        \< fromIntegral c, c'v4l2_control'value = toControlData x }) ==<< zero
175

unfoldM :: Monad m => (b -> m (Maybe (a, b))) -> b -> m [a]
unfoldM f b = do
        fb <- f b
180   case fb of
        Just (a, b') -> (a :) 'liftM' unfoldM f b'
        Nothing -> return []

```

## 9 Graphics/V4L2/Device.hs

```

{-# LANGUAGE DeriveDataTypeable, GeneralizedNewtypeDeriving #-}
{- |
Module      : Graphics.V4L2.Device
Maintainer  : claude@mathr.co.uk
5   Stability : no
   Portability : no
-}
module Graphics.V4L2.Device
  ( Device()

```

```

10     , openDevice
      , closeDevice
      , withDevice
    ) where

15 import Control.Exception (bracket)
import Data.Bits (Bits)
import Data.Typeable (Typeable)
import Foreign.C.Error (throwErrnoIfMinus1, throwErrnoIfMinus1_)
import Foreign.C.String (withCString)
20 import Foreign.Storable (Storable)
import System.Posix.Types (Fd)

import Bindings.LibV4L2 (c'v4l2_open, c'v4l2_close)
import Bindings.Posix.Fcntl (c'ORDWR)

25 {- | Device handle. -}
newtype Device = Device Fd
    deriving (Bits, Bounded, Enum, Eq, Integral, Num, Ord, Read, Real, Show, ↵
            ↵ Storable, Typeable)

30 {- | Open a device.
      Fails with invalid argument when the device is not a V4L2 device.
    -}
openDevice :: FilePath {- ^ device name -} -> IO Device
openDevice f = withCString f $ \s -> do
35     h <- throwErrnoIfMinus1 "Graphics.V4L2.Device.openDevice" (c'v4l2_open s c'↵
            ↵ ORDWR 0)
    return (fromIntegral h)

{- | Close a device. -}
closeDevice :: Device {- ^ device handle -} -> IO ()
40 closeDevice d = throwErrnoIfMinus1_ "Graphics.V4L2.Device.closeDevice" (c'↵
            ↵ v4l2_close (fromIntegral d))

{- | Perform an action with a device.
      The device will be close on exit from withDevice, whether by
      normal termination or by raising an exception. If closing the
45     device raises an exception, then this exception will be raised by
      'withDevice' rather than any exception raised by the action.
    -}
withDevice :: FilePath {- ^ device name -} -> (Device -> IO a) {- ^ action -} ->↵
            ↵ IO a
withDevice f = bracket (openDevice f) closeDevice

```

## 10 Graphics/V4L2/Field.hs

```

{- |
Module      : Graphics.V4L2.Field
Maintainer  : claude@mathr.co.uk
Stability   : no
5  Portability : no
-}
module Graphics.V4L2.Field
    ( module Graphics.V4L2.Field.Internal
    ) where
10

```

```
import Graphics.V4L2.Field.Internal hiding (fromField, toField)
```

## 11 Graphics/V4L2/Field/Internal.hs

```
{-# LANGUAGE DeriveDataTypeable #-}
{- |
Module      : Graphics.V4L2.Field.Internal
Maintainer  : claudem@mathr.co.uk
5   Stability : no
   Portability : no
-}
module Graphics.V4L2.Field.Internal
  ( Field(..)
10   , fieldHasTop
    , fieldHasBottom
    , fieldHasBoth
    , fromField
    , toField
15   ) where

import Data.Data (Data)
import Data.Typeable (Typeable)
import Data.Word (Word32)
20 import System.IO.Unsafe (unsafePerformIO)
import Foreign.Marshal.Utils (toBool)

import Bindings.Linux.VideoDev2

25 import Foreign.Extra.CEnum (toCEnum, fromCEnum)

{- | Interlacing modes. -}
data Field
  = FieldAny
30   | FieldNone
   | FieldTop
   | FieldBottom
   | FieldInterlaced
   | FieldSeqTB
35   | FieldSeqBT
   | FieldAlternate
   | FieldInterlacedTB
   | FieldInterlacedBT
   | FieldUnknown Word32
40   deriving (Eq, Ord, Read, Show, Data, Typeable)

{- | internal -}
fromField :: C'v4l2_field -> Field
{- | internal -}
45 toField :: Field -> C'v4l2_field

(fromField, toField) = (fromCEnum spec (FieldUnknown . fromIntegral), toCEnum ↯
  ↯ spec isUnknown unUnknown) where
  spec =
50   [ ( FieldAny           , c'V4L2_FIELD_ANY           )
     , ( FieldNone        , c'V4L2_FIELD_NONE          )
     , ( FieldTop         , c'V4L2_FIELD_TOP            )
     , ( FieldBottom      , c'V4L2_FIELD_BOTTOM        )
```

```

    , ( FieldInterlaced    , c'V4L2.FIELD.INTERLACED    )
    , ( FieldSeqTB         , c'V4L2.FIELD.SEQ_TB         )
55   , ( FieldSeqBT        , c'V4L2.FIELD.SEQ_BT        )
    , ( FieldAlternate     , c'V4L2.FIELD.ALTERNATE     )
    , ( FieldInterlacedTB  , c'V4L2.FIELD.INTERLACED_TB )
    , ( FieldInterlacedBT  , c'V4L2.FIELD.INTERLACED_BT )
    ]
60   isUnknown (FieldUnknown _) = True
    isUnknown _ = False
    unUnknown (FieldUnknown f) = fromIntegral f
    unUnknown _ = error "Graphice.V4L2.Field.Internal.toField"

65   {- | Inspect field interlacing. -}
    fieldHasTop :: Field -> Bool
    fieldHasTop f = unsafePerformIO $ toBool 'fmap' c'V4L2.FIELD.HAS_TOP    (↯
        ↪ toField f)

    {- | Inspect field interlacing. -}
70   fieldHasBottom :: Field -> Bool
    fieldHasBottom f = unsafePerformIO $ toBool 'fmap' c'V4L2.FIELD.HAS_BOTTOM (↯
        ↪ toField f)

    {- | Inspect field interlacing. -}
    fieldHasBoth :: Field -> Bool
75   fieldHasBoth f = unsafePerformIO $ toBool 'fmap' c'V4L2.FIELD.HAS_BOTH    (↯
        ↪ toField f)

```

## 12 Graphics/V4L2/Format.hs

```

{-# LANGUAGE DeriveDataTypeable , GeneralizedNewtypeDeriving #-}
{- |
Module      : Graphics.V4L2.Format
Maintainer  : claude@mathr.co.uk
5   Stability : no
   Portability : no
-}
module Graphics.V4L2.Format
  ( BufferType(..)
10   , Direction(..)
    , Format(..)
    , FormatID()
    , FormatDescription(..)
    , FormatFlag(..)
15   , queryFormats
    , FrameSize(..)
    , FrameSizes(..)
    , queryFrameSizes
    , FrameIntervals(..)
20   , queryFrameIntervals
    , ImageFormat(..)
    ) where

import Control.Exception as E (catch , throwIO)
import Control.Monad (when)
import Data.Data (Data)
import Data.Map (Map)
import qualified Data.Map as M

```

```

import Data.Set (Set)
30 import qualified Data.Set as S
import Data.Typeable (Typeable)
import Data.Word (Word32)
import GHC.IO.Exception (IOErrorType(InvalidArgument), ioe_type)

35 import Bindings.Linux.VideoDev2

import Foreign.Extra.BitSet (fromBitSet)
import Foreign.Extra.CEnum (toCEnum)
import Foreign.Extra.String (fromString)
40 import Graphics.V4L2.Device (Device)
import Graphics.V4L2.IOctl (ioctl, zero)
import Graphics.V4L2.Field (Field)
import Graphics.V4L2.Field.Internal (fromField, toField)
import Graphics.V4L2.PixelFormat (PixelFormat)
45 import Graphics.V4L2.PixelFormat.Internal (fromPixelFormat, toPixelFormat)
import Graphics.V4L2.ColorSpace (ColorSpace)
import Graphics.V4L2.ColorSpace.Internal (fromColorSpace, toColorSpace)
import Graphics.V4L2.Types (Fraction(..))
import Graphics.V4L2.Types.Internal (fromFraction)

50 -- FIXME {
  {-
  c'V4L2_BUF_TYPE_VIDEO_CAPTURE_MPLANE = 9
  c'V4L2_BUF_TYPE_VIDEO_OUTPUT_MPLANE = 10
55 V4L2_BUF_TYPE_VIDEO_OVERLAY
  V4L2_BUF_TYPE_VBI_CAPTURE
  V4L2_BUF_TYPE_VBI_OUTPUT
  V4L2_BUF_TYPE_SLICED_VBI_CAPTURE
  V4L2_BUF_TYPE_SLICED_VBI_OUTPUT
60 V4L2_BUF_TYPE_VIDEO_OUTPUT_OVERLAY
  V4L2_BUF_TYPE_PRIVATE
  -}
  -- FIXME }

65 {- | Buffer types. -}
data BufferType = BufferVideoCapture | BufferVideoOutput | BufferUnknown Word32
  deriving (Eq, Ord, Read, Show, Data, Typeable)

toBufferType :: BufferType -> C'v4l2_buf_type
70 toBufferType = toCEnum
  [ ( BufferVideoCapture , c'V4L2_BUF_TYPE_VIDEO_CAPTURE )
  , ( BufferVideoOutput   , c'V4L2_BUF_TYPE_VIDEO_OUTPUT   )
  ] isU unU
  where
75   isU (BufferUnknown _) = True
   isU _ = False
   unU (BufferUnknown b) = fromIntegral b
   unU _ = error "err"

80 {- | Transfer types. -}
data Direction = Capture | Output
  deriving (Eq, Ord, Enum, Bounded, Read, Show, Data, Typeable)

{- | Buffer formats. -}
85 class Format f where

```

```

{- | Corresponding buffer type. -}
formatBufferType :: f -> Direction -> BufferType
{- | Query the format. -}
getFormat :: Device -> Direction -> IO f
90 {- | Select the format. -}
setFormat :: Device -> Direction -> f -> IO f
{- | Test the format. -}
tryFormat :: Device -> Direction -> f -> IO f

95 {- | Image format. -}
data ImageFormat = ImageFormat
  { imageWidth, imageHeight :: Int
  , imagePixelFormat :: PixelFormat
  , imageField :: Field
100 , imageBytesPerLine :: Int
  , imageSize :: Int
  , imageColorSpace :: ColorSpace
  }
  deriving (Eq, Ord, Read, Show, Data, Typeable)

105 instance Format ImageFormat where
  formatBufferType _ = imageBufferType
  getFormat d dir =
    return . decodeImageFormat . c'v4l2_format_u'pix . c'v4l2_format'fmt <<< \
      \> iocctl d C'VIDIOC_G_FMT . (\s->s{ c'v4l2_format'type = toBufferType $ \
      \> imageBufferType dir }) <<< zero
110 setFormat d dir f = do
  u <- ('u'v4l2_format_u'pix' encodeImageFormat f) <<< zero
  return . decodeImageFormat . c'v4l2_format_u'pix . c'v4l2_format'fmt <<< \
    \> iocctl d C'VIDIOC_S_FMT <<< (\s->s{ c'v4l2_format'type = toBufferType $ \
    \> imageBufferType dir, c'v4l2_format'fmt = u }) 'fmap' zero
  tryFormat d dir f = do
    u <- ('u'v4l2_format_u'pix' encodeImageFormat f) <<< zero
115 return . decodeImageFormat . c'v4l2_format_u'pix . c'v4l2_format'fmt <<< \
  \> iocctl d C'VIDIOC_TRY_FMT <<< (\s->s{ c'v4l2_format'type = toBufferType \
  \> $ imageBufferType dir, c'v4l2_format'fmt = u }) 'fmap' zero

encodeImageFormat :: ImageFormat -> C'v4l2_pix_format
encodeImageFormat f = C'v4l2_pix_format
  { c'v4l2_pix_format'width = fromIntegral $ imageWidth f
120 , c'v4l2_pix_format'height = fromIntegral $ imageHeight f
  , c'v4l2_pix_format'pixelformat = toPixelFormat $ imagePixelFormat f
  , c'v4l2_pix_format'field = toField $ imageField f
  , c'v4l2_pix_format'bytesperline = fromIntegral $ imageBytesPerLine f
  , c'v4l2_pix_format'sizeimage = fromIntegral $ imageSize f
125 , c'v4l2_pix_format'colospace = toColorSpace $ imageColorSpace f
  , c'v4l2_pix_format'priv = 0
  }

decodeImageFormat :: C'v4l2_pix_format -> ImageFormat
130 decodeImageFormat f = ImageFormat
  { imageWidth = fromIntegral $ c'v4l2_pix_format'width f
  , imageHeight = fromIntegral $ c'v4l2_pix_format'height f
  , imagePixelFormat = fromPixelFormat $ c'v4l2_pix_format'pixelformat f
  , imageField = fromField $ c'v4l2_pix_format'field f
135 , imageBytesPerLine = fromIntegral $ c'v4l2_pix_format'bytesperline f
  , imageSize = fromIntegral $ c'v4l2_pix_format'sizeimage f

```

```

    , imageColorSpace = fromColorSpace $ c'v4l2_pix_format' colorspace f
  }

140 imageBufferType :: Direction -> BufferType
imageBufferType Capture = BufferVideoCapture
imageBufferType Output = BufferVideoOutput

145 {- | Format flags. -}
data FormatFlag
    = FormatCompressed
    | FormatEmulated
    | FormatUnknown Word32
150 deriving (Eq, Ord, Read, Show, Data, Typeable)

fromFormatFlag :: Word32 -> Set FormatFlag
fromFormatFlag = fromBitSet
  [ ( FormatCompressed , c'V4L2_FMT_FLAG_COMPRESSED )
155   , ( FormatEmulated   , c'V4L2_FMT_FLAG_EMULATED   )
    ] FormatUnknown

{- | Video format ID. -}
newtype FormatID = FormatID Int
160 deriving (Eq, Ord, Enum, Bounded, Read, Show, Data, Typeable, Num, Integral, <
    < Real)

{- | Video format description -}
data FormatDescription = FormatDescription
  { formatDescription :: String
165   , formatPixelFormat :: PixelFormat
    , formatFlags :: Set FormatFlag
  }
deriving (Eq, Ord, Read, Show, Data, Typeable)

170 enumfmt :: Device -> BufferType -> FormatID -> IO FormatDescription
enumfmt h t n = do
  d <- ioctl h C'VIDIOC_ENUM_FMT . (\s->s{ c'v4l2_fmtdesc'index = fromIntegral n<
    < , c'v4l2_fmtdesc'type = toBufferType t }) =<< zero
  return FormatDescription
    { formatDescription = fromString $ c'v4l2_fmtdesc'description d
175    , formatPixelFormat = fromPixelFormat $ c'v4l2_fmtdesc'pixelformat d
    , formatFlags = fromFormatFlag $ c'v4l2_fmtdesc'flags d
    }

{- | Enumerate supported buffer formats.
180
    Exceptions:

        * InvalidArgument - buffer type not supported or index out of range
    -}
185 queryFormats :: Device -> BufferType -> IO (Map FormatID FormatDescription)
queryFormats = enumfmts' 0

enumfmts' :: FormatID -> Device -> BufferType -> IO (Map FormatID <
    < FormatDescription)
enumfmts' n h t = do
190   mi <- (Just 'fmap' enumfmt h t n) 'E.catch' (\e -> case ioe_type e of

```

```

    InvalidArgument -> return Nothing
    _ -> throwIO e)
case mi of
    Just i -> M.insert n i 'fmap' enumfmts' (n + 1) h t
195    Nothing -> return M.empty

{- | Enumerate supported frame sizes. -}
queryFrameSizes :: Device -> PixelFormat -> IO FrameSizes
queryFrameSizes h p = do
200    fs <- enumframesizes0 h p 'E.catch' (\e -> case ioe_type e of
        InvalidArgument -> return (DiscreteSizes S.empty)
        _ -> throwIO e)
    case fs of
        DiscreteSizes f | not $ S.null f -> do
205            fs' <- enumframesizesD 1 h p
            return . DiscreteSizes $ f 'S.union' fs'
        _ -> return fs

enumframesizes0 :: Device -> PixelFormat -> IO FrameSizes
210 enumframesizes0 h p = do
    f <- ioctl h C'VIDIOC_ENUM_FRAMESIZES . (\s->s{ c'v4l2_frmsizeenum'index = 0, ↵
        ↵ c'v4l2_frmsizeenum'pixel_format = toPixelFormat p }) ==<< zero
    case c'v4l2_frmsizeenum'type f of
        x | x == c'V4L2_FRMSIZE_TYPE_DISCRETE -> return . DiscreteSizes . S.↵
            ↵ singleton . decodeFrameSize . c'v4l2_frmsizeenum_u'discrete . c'↵
            ↵ v4l2_frmsizeenum'u $ f
        | x == c'V4L2_FRMSIZE_TYPE_STEPWISE ||
215        x == c'V4L2_FRMSIZE_TYPE_CONTINUOUS -> do
            let s = c'v4l2_frmsizeenum_u'steppwise . c'v4l2_frmsizeenum'u $ f
            return StepwiseSizes
                { stepwiseMinWidth = fromIntegral $ c'v4l2_frmsize_stepwise'↵
                    ↵ min_width s
                , stepwiseMaxWidth = fromIntegral $ c'v4l2_frmsize_stepwise'↵
                    ↵ max_width s
220                , stepwiseStepWidth = fromIntegral $ c'v4l2_frmsize_stepwise'↵
                    ↵ step_width s
                , stepwiseMinHeight = fromIntegral $ c'v4l2_frmsize_stepwise'↵
                    ↵ min_height s
                , stepwiseMaxHeight = fromIntegral $ c'v4l2_frmsize_stepwise'↵
                    ↵ max_height s
                , stepwiseStepHeight = fromIntegral $ c'v4l2_frmsize_stepwise'↵
                    ↵ step_height s
                }
225        | otherwise -> error "err"

enumframesizesD :: Word32 -> Device -> PixelFormat -> IO (Set FrameSize)
enumframesizesD n h p = do
    f <- (do
230        f' <- ioctl h C'VIDIOC_ENUM_FRAMESIZES . (\s->s{ c'v4l2_frmsizeenum'index = ↵
            ↵ n, c'v4l2_frmsizeenum'pixel_format = toPixelFormat p }) ==<< zero
        when (c'v4l2_frmsizeenum'type f' /= c'V4L2_FRMSIZE_TYPE_DISCRETE) $ error "↵
            ↵ err"
        return (Just f')) 'E.catch' (\e -> case ioe_type e of
            InvalidArgument -> return Nothing
            _ -> throwIO e)
235    case f of
        Nothing -> return S.empty

```

```

    Just fs -> S.insert (decodeFrameSize . c'v4l2_frmsizeenum_u' discrete . c'↵
        ↵ v4l2_frmsizeenum'u $ fs) 'fmap' enumframesizesD (n + 1) h p

{- | Single frame size. -}
240 data FrameSize = FrameSize
    { frameWidth
    , frameHeight :: Int
    }
    deriving (Eq, Ord, Read, Show, Data, Typeable)

245 {- | Discrete and continuous frame sizes. -}
data FrameSizes
    = DiscreteSizes
    { discreteSizes :: Set FrameSize }
250 | StepwiseSizes
    { stepwiseMinWidth
    , stepwiseMaxWidth
    , stepwiseStepWidth
    , stepwiseMinHeight
255 , stepwiseMaxHeight
    , stepwiseStepHeight :: Int
    }
    deriving (Eq, Ord, Read, Show, Data, Typeable)

260 decodeFrameSize :: C'v4l2_frmsize_discrete -> FrameSize
decodeFrameSize d = FrameSize
    { frameWidth = fromIntegral $ c'v4l2_frmsize_discrete'width d
    , frameHeight = fromIntegral $ c'v4l2_frmsize_discrete'height d
    }

265 {- | Enumerate frame intervals. -}
queryFrameIntervals :: Device -> PixelFormat -> FrameSize -> IO FrameIntervals
queryFrameIntervals h p f = do
    fi <- enumframeintervals0 h p f 'E.catch' (\e -> case ioe_type e of
270     InvalidArgument -> return (DiscreteIntervals S.empty)
    _ -> throwIO e)
    case fi of
        DiscreteIntervals i | not $ S.null i -> do
            is <- enumframeintervalsD 1 h p f
275     return . DiscreteIntervals $ i 'S.union' is
        _ -> return fi

enumframeintervals0 :: Device -> PixelFormat -> FrameSize -> IO FrameIntervals
enumframeintervals0 h p f = do
280     i <- ioctl h C'VIDIOC_ENUMFRAMEINTERVALS . (\s->s{ c'v4l2_frmivalenum'index =↵
        ↵ 0, c'v4l2_frmivalenum'pixel_format = toPixelFormat p, c'↵
        ↵ v4l2_frmivalenum'width = fromIntegral (frameWidth f), c'v4l2_frmivalenum'↵
        ↵ 'height = fromIntegral (frameHeight f) }) ==<< zero
    case c'v4l2_frmivalenum'type i of
        x | x == c'V4L2_FRMIVAL_TYPE_DISCRETE -> return . DiscreteIntervals . S.↵
            ↵ singleton . fromFraction . c'v4l2_frmivalenum_u' discrete . c'↵
            ↵ v4l2_frmivalenum'u $ i
        | x == c'V4L2_FRMIVAL_TYPE_STEPWISE ||
285     x == c'V4L2_FRMIVAL_TYPE_CONTINUOUS -> do
            let s = c'v4l2_frmivalenum_u'steppwise . c'v4l2_frmivalenum'u $ i
            return StepwiseIntervals
                { stepwiseMin = fromFraction $ c'v4l2_frmival_stepwise'min s

```

```

        , stepwiseMax = fromFraction $ c'v4l2_frmival_stepwise'max s
        , stepwiseStep = fromFraction $ c'v4l2_frmival_stepwise'step s
290     }
        | otherwise -> error "err"

enumframeintervalsD :: Word32 -> Device -> PixelFormat -> FrameSize -> IO (Set ↵
    ↵ Fraction)
enumframeintervalsD n h p f = do
295   i <- (do
        f' <- ioctl h C'VIDIOC_ENUMFRAMEINTERVALS . (\s->s{ c'v4l2_frmivalenum'↵
            ↵ index = n, c'v4l2_frmivalenum'pixel_format = toPixelFormat p, c'↵
            ↵ v4l2_frmivalenum'width = fromIntegral (frameWidth f), c'↵
            ↵ v4l2_frmivalenum'height = fromIntegral (frameHeight f) }) =<< zero
        when (c'v4l2_frmivalenum'type f' /= c'V4L2_FRMIVAL_TYPE_DISCRETE) $ error "↵
            ↵ err"
        return (Just f')) 'E.catch' (\e -> case ioe_type e of
300     InvalidArgument -> return Nothing
        _ -> throwIO e)
    case i of
        Nothing -> return S.empty
        Just fs -> S.insert (fromFraction . c'v4l2_frmivalenum_u'discrete . c'↵
            ↵ v4l2_frmivalenum'u $ fs) 'fmap' enumframeintervalsD (n + 1) h p f

305 {- | Discrete and continuous frame intervals. -}
data FrameIntervals
    = DiscreteIntervals
        { discreteIntervals :: Set Fraction }
    | StepwiseIntervals
310     { stepwiseMin, stepwiseMax, stepwiseStep :: Fraction }
    deriving (Eq, Ord, Read, Show, Data, Typeable)

```

## 13 Graphics/V4L2.hs

```

{- |
Module      : Graphics.V4L2
Maintainer  : claude@mathr.co.uk
Stability   : no
5  Portability : no
-}

module Graphics.V4L2 (
    module Graphics.V4L2.Device
    , module Graphics.V4L2.Capability
10   , module Graphics.V4L2.Priority
    , module Graphics.V4L2.Control
    , module Graphics.V4L2.Field
    , module Graphics.V4L2.PixelFormat
    , module Graphics.V4L2.ColorSpace
15   , module Graphics.V4L2.Format
    , module Graphics.V4L2.VideoInput
    , module Graphics.V4L2.VideoStandard
    , module Graphics.V4L2.VideoCapture
    ) where

20
import Graphics.V4L2.Device
import Graphics.V4L2.Capability
import Graphics.V4L2.Priority
import Graphics.V4L2.Control

```

```

25  import Graphics.V4L2.Field
    import Graphics.V4L2.PixelFormat
    import Graphics.V4L2.ColorSpace
    import Graphics.V4L2.Format
    import Graphics.V4L2.VideoInput
30  import Graphics.V4L2.VideoStandard
    import Graphics.V4L2.VideoCapture

    {-
      -- * Buffers and formats.
35      , BufferType(..)
      -- ** Image formats.
      -- * Audio inputs.
      , AudioIndex
      -- * Tuners.
40      , TunerIndex
      -- * Exceptions
      , V4L2Bug(..)
    ) where
import Prelude hiding (null, catch)
45  import Foreign
    import Foreign.C
    import Foreign.Helper
    import Control.Monad (when)
    import Control.Exception
50  import GHC.IO.Exception (IOErrorType(InvalidArgument, ResourceBusy), ioe_type)
    import System.Posix.Types (Fd(Fd), CMode)
    import System.Posix.IOctl (IOControl(ioctlReq))
    import Data.Set (Set, empty, fromList, singleton, union, insert, null)
    import Data.Word (Word8, Word32, Word64)
55  import Data.Bits (shiftR, (.&.), testBit)
    import Data.Maybe (fromMaybe)
    import Bindings.Linux.VideoDev2
    import Bindings.LibV4L2

60  {- | Buffer type identifiers. -}
    enum "BufferType" "C'v4l2_buf_type" "c'V4L2_BUF_TYPE_" "B" $ words "↵
      ↵ VIDEO_CAPTURE VIDEO_CAPTURE_MPLANE VIDEO_OUTPUT VIDEO_OUTPUT_MPLANE ↵
      ↵ VIDEO_OVERLAY VBL_CAPTURE VBL_OUTPUT SLICED_VBL_CAPTURE SLICED_VBL_OUTPUT ↵
      ↵ VIDEO_OUTPUT_OVERLAY"

    {- | Audio input index. -}
    newtype AudioIndex = AudioIndex Int
65      deriving (Eq, Ord, Enum, Bounded, Read, Show, Data, Typeable, Num, Integral, ↵
        ↵ Real)

    {- | Tuner index. -}
    newtype TunerIndex = TunerIndex Int
      deriving (Eq, Ord, Enum, Bounded, Read, Show, Data, Typeable, Num, Integral, ↵
        ↵ Real)
70

    zero :: Storable a => IO a
    zero = alloca $ \p -> c'memset p 0 (fromIntegral $ sizeOf (undefined 'asTypeOf' ↵
      ↵ unsafePerformIO (peek p))) >> peek p

75  {- | Internal errors. Please report these as bugs. -}

```

```

data V4L2Bug
  = V4L2BugUnparseableInput C'v4l2_input
  | V4L2BugUnparseableFormat C'v4l2_format
  | V4L2BugUnparseableFormatDescription C'v4l2_fmtdesc
80  | V4L2BugUnexpectedFrameSizeType C'v4l2_frmsizeenum
  | V4L2BugUnexpectedFrameIntervalType C'v4l2_frmivalenum
  deriving (Show, Typeable)
instance Exception V4L2Bug
-}

```

## 14 Graphics/V4L2/IOCtl.hs

```

{-# LANGUAGE ForeignFunctionInterface #-}
{- |
Module      : Graphics.V4L2.IOCtl
Maintainer  : claud@mathr.co.uk
5  Stability : no
Portability : no

Heavily based on the 'ioctl' package (c) Maciej Piechotka.

10 -}
module Graphics.V4L2.IOCtl
  ( ioctl
    , ioctl_
    , ioctl'
15  , zero
    ) where

import Foreign (Ptr, Storable, alloca, castPtr, peek, sizeOf, with)
import Foreign.C (CInt(..), CSize(..), throwErrnoIfMinus1_)
20 import System.Posix.IOCtl (IOControl(ioctlReq))
import System.IO.Unsafe (unsafePerformIO)

import Bindings.LibV4L2 (c'v4l2_ioctl)

25 import Graphics.V4L2.Device (Device)

c_ioctl' :: IOControl req d => Device -> req -> Ptr d -> IO ()
c_ioctl' f req p =
  throwErrnoIfMinus1_ "ioctl" $
30   c'v4l2_ioctl (fromIntegral f) (fromIntegral $ ioctlReq req) (castPtr p)

-- | Calls a ioctl reading the structure after the call
ioctl :: IOControl req d
  => Device -- ^ The file descriptor
  -> req -- ^ The request
35  -> d -- ^ The data
  -> IO d -- ^ The data after the call
ioctl f req d = with d $ \p -> c_ioctl' f req p >> peek p

40 -- | Call a ioctl ignoring the result
ioctl_ :: IOControl req d
  => Device -- ^ The file descriptor
  -> req -- ^ The request
  -> d -- ^ The data
45  -> IO ()

```

```

ioctl_ f req d = with d $ \p -> c_ioctl' f req p

-- | Call a ioctl with uninitialized data
ioctl' :: IOControl req d
50   => Device -- ^ The file descriptor
    -> req -- ^ The request
    -> IO d -- ^ The data
ioctl' f req = alloca $ \p -> c_ioctl' f req p >> peek p

55 {- | A value obtained by peeking cleared memory. -}
zero :: Storable a => IO a
zero = alloca $ \p -> c'memset p 0 (fromIntegral $ sizeof (undefined 'asTypeOf' ↯
    ↯ unsafePerformIO (peek p))) >> peek p

foreign import ccall "string.h memset" c'memset :: Ptr a -> CInt -> CSize -> IO ↯
    ↯ (Ptr a)

```

## 15 Graphics/V4L2/PixelFormat.hs

```

{- |
Module      : Graphics.V4L2.PixelFormat
Maintainer  : claude@mathr.co.uk
Stability   : no
5  Portability : no
-}
module Graphics.V4L2.PixelFormat
  ( module Graphics.V4L2.PixelFormat.Internal
  ) where

10 import Graphics.V4L2.PixelFormat.Internal hiding (fromPixelFormat, toPixelFormat ↯
    ↯ )

```

## 16 Graphics/V4L2/PixelFormat/Internal.hs

```

{-# LANGUAGE DeriveDataTypeable #-}
{- |
Module      : Graphics.V4L2.PixelFormat.Internal
Maintainer  : claude@mathr.co.uk
5  Stability   : no
  Portability : no
-}
module Graphics.V4L2.PixelFormat.Internal
  ( PixelFormat(..)
10   , fromPixelFormat
    , toPixelFormat
    ) where

import Data.Data (Data)
15 import Data.Typeable (Typeable)
import Data.Word (Word32)

import Bindings.Linux.VideoDev2

20 import Foreign.Extra.CEnum (toCEnum, fromCEnum)

{- | Pixel formats. -}

```

```

data PixelFormat
  -- grep PIX /usr/include/linux/videodev2.h | sed "s|^#define \((V4L2_PIX_FMT_ℵ
    ↳ \([^\t]*\)\)\([ \t]*.*\)| \[ \t\] Pixel\2|"
25  = PixelRGB332
    | PixelRGB444
    | PixelRGB555
    | PixelRGB565
    | PixelRGB555X
30  | PixelRGB565X
    | PixelBGR24
    | PixelRGB24
    | PixelBGR32
    | PixelRGB32
35  | PixelGREY
    | PixelY16
    | PixelPAL8
    | PixelYVU410
    | PixelYVU420
40  | PixelYUYV
    | PixelYYUV
    | PixelYVYU
    | PixelUYVY
    | PixelVYUY
45  | PixelYUV422P
    | PixelYUV411P
    | PixelY41P
    | PixelYUV444
    | PixelYUV555
50  | PixelYUV565
    | PixelYUV32
    | PixelYUV410
    | PixelYUV420
    | PixelHI240
55  | PixelHM12
    | PixelNV12
    | PixelNV21
    | PixelNV16
    | PixelNV61
60  | PixelSBGGR8
    | PixelSGBRG8
    | PixelSGRBG8
    | PixelSGRBG10
    | PixelSGRBG10DPCM8
65  | PixelSBGGR16
    | PixelMJPEG
    | PixelJPEG
    | PixelDV
    | PixelMPEG
70  | PixelWNVA
    | PixelSN9C10X
    | PixelSN9C20X_I420
    | PixelPWC1
    | PixelPWC2
75  | PixelET61X251
    | PixelSPCA501
    | PixelSPCA505
    | PixelSPCA508

```

```

    | PixelSPCA561
80  | PixelPAC207
    | PixelMR97310A
    | PixelSQ905C
    | PixelPJPEG
    | PixelOV511
85  | PixelOV518
    | PixelUnknown Word32
    deriving (Eq, Ord, Read, Show, Data, Typeable)

{- | internal -}
90 fromPixelFormat :: Word32 -> PixelFormat
{- | internal -}
toPixelFormat :: PixelFormat -> Word32

(fromPixelFormat, toPixelFormat) = (fromCEnum spec PixelUnknown, toCEnum spec ↯
    ↯ isUnknown unUnknown) where
95  -- grep PIX /usr/include/linux/videodev2.h | sed "s|^#define \((V4L2_PIX_FMT_↯
    ↯ \([^\t]*\) \)[ \t].*$$" , ( Pixel\2 , c'\1 )|"
    spec =
        [ ( PixelRGB332 , c'V4L2_PIX_FMT_RGB332 )
          , ( PixelRGB444 , c'V4L2_PIX_FMT_RGB444 )
          , ( PixelRGB555 , c'V4L2_PIX_FMT_RGB555 )
100  , ( PixelRGB565 , c'V4L2_PIX_FMT_RGB565 )
          , ( PixelRGB555X , c'V4L2_PIX_FMT_RGB555X )
          , ( PixelRGB565X , c'V4L2_PIX_FMT_RGB565X )
          , ( PixelBGR24 , c'V4L2_PIX_FMT_BGR24 )
          , ( PixelRGB24 , c'V4L2_PIX_FMT_RGB24 )
105  , ( PixelBGR32 , c'V4L2_PIX_FMT_BGR32 )
          , ( PixelRGB32 , c'V4L2_PIX_FMT_RGB32 )
          , ( PixelGREY , c'V4L2_PIX_FMT_GREY )
          , ( PixelY16 , c'V4L2_PIX_FMT_Y16 )
          , ( PixelPAL8 , c'V4L2_PIX_FMT_PAL8 )
110  , ( PixelYVU410 , c'V4L2_PIX_FMT_YVU410 )
          , ( PixelYVU420 , c'V4L2_PIX_FMT_YVU420 )
          , ( PixelYUYV , c'V4L2_PIX_FMT_YUYV )
          , ( PixelYYUV , c'V4L2_PIX_FMT_YYUV )
          , ( PixelYVYU , c'V4L2_PIX_FMT_YVYU )
115  , ( PixelUYVY , c'V4L2_PIX_FMT_UYVY )
          , ( PixelVYUY , c'V4L2_PIX_FMT_VYUY )
          , ( PixelYUV422P , c'V4L2_PIX_FMT_YUV422P )
          , ( PixelYUV411P , c'V4L2_PIX_FMT_YUV411P )
          , ( PixelY41P , c'V4L2_PIX_FMT_Y41P )
120  , ( PixelYUV444 , c'V4L2_PIX_FMT_YUV444 )
          , ( PixelYUV555 , c'V4L2_PIX_FMT_YUV555 )
          , ( PixelYUV565 , c'V4L2_PIX_FMT_YUV565 )
          , ( PixelYUV32 , c'V4L2_PIX_FMT_YUV32 )
          , ( PixelYUV410 , c'V4L2_PIX_FMT_YUV410 )
125  , ( PixelYUV420 , c'V4L2_PIX_FMT_YUV420 )
          , ( PixelHI240 , c'V4L2_PIX_FMT_HI240 )
          , ( PixelHM12 , c'V4L2_PIX_FMT_HM12 )
          , ( PixelNV12 , c'V4L2_PIX_FMT_NV12 )
          , ( PixelNV21 , c'V4L2_PIX_FMT_NV21 )
130  , ( PixelNV16 , c'V4L2_PIX_FMT_NV16 )
          , ( PixelNV61 , c'V4L2_PIX_FMT_NV61 )
          , ( PixelSBGGR8 , c'V4L2_PIX_FMT_SBGGR8 )
          , ( PixelSGBRG8 , c'V4L2_PIX_FMT_SGBRG8 )

```

```

135     , ( PixelSGBG8 , c'V4L2_PIX_FMT_SGBG8 )
      , ( PixelSGBG10 , c'V4L2_PIX_FMT_SGBG10 )
      , ( PixelSGBG10DPCM8 , c'V4L2_PIX_FMT_SGBG10DPCM8 )
      , ( PixelSBGGR16 , c'V4L2_PIX_FMT_SBGGR16 )
      , ( PixelMJPEG , c'V4L2_PIX_FMT_MJPEG )
      , ( PixelJPEG , c'V4L2_PIX_FMT_JPEG )
140     , ( PixelDV , c'V4L2_PIX_FMT_DV )
      , ( PixelMPEG , c'V4L2_PIX_FMT_MPEG )
      , ( PixelWNVA , c'V4L2_PIX_FMT_WNVA )
      , ( PixelSN9C10X , c'V4L2_PIX_FMT_SN9C10X )
      , ( PixelSN9C20X_I420 , c'V4L2_PIX_FMT_SN9C20X_I420 )
145     , ( PixelPWC1 , c'V4L2_PIX_FMT_PWC1 )
      , ( PixelPWC2 , c'V4L2_PIX_FMT_PWC2 )
      , ( PixelET61X251 , c'V4L2_PIX_FMT_ET61X251 )
      , ( PixelSPCA501 , c'V4L2_PIX_FMT_SPCA501 )
      , ( PixelSPCA505 , c'V4L2_PIX_FMT_SPCA505 )
150     , ( PixelSPCA508 , c'V4L2_PIX_FMT_SPCA508 )
      , ( PixelSPCA561 , c'V4L2_PIX_FMT_SPCA561 )
      , ( PixelPAC207 , c'V4L2_PIX_FMT_PAC207 )
      , ( PixelMR97310A , c'V4L2_PIX_FMT_MR97310A )
      , ( PixelSQ905C , c'V4L2_PIX_FMT_SQ905C )
155     , ( PixelPJPG , c'V4L2_PIX_FMT_PJPG )
      , ( PixelOV511 , c'V4L2_PIX_FMT_OV511 )
      , ( PixelOV518 , c'V4L2_PIX_FMT_OV518 )
    ]
    isUnknown (PixelUnknown _) = True
160    isUnknown _ = False
    unUnknown (PixelUnknown f) = f
    unUnknown _ = error "Graphice.V4L2.PixelFormat.Internal.toPixelFormat"

```

## 17 Graphics/V4L2/Priority.hs

```

{-# LANGUAGE DeriveDataTypeable #-}
{- |
Module      : Graphics.V4L2.Priority
Maintainer  : claude@mathr.co.uk
5  Stability : no
Portability : no
-}
module Graphics.V4L2.Priority
  ( Priority(..)
10   , getPriority
    , setPriority
    ) where

import Data.Data (Data)
15 import Data.Typeable (Typeable)
import Data.Word (Word32)

import Bindings.Linux.VideoDev2

20 import Foreign.Extra.CEnum (fromCEnum, toCEnum)
import Graphics.V4L2.Device (Device)
import Graphics.V4L2.IOctl (ioctl', ioctl_)

{- | Priorities. -}
25 data Priority

```

```

    = PriorityUnset
    | PriorityBackground
    | PriorityInteractive
    | PriorityDefault
30  | PriorityRecord
    | PriorityUnknown Word32
    deriving (Eq, Ord, Read, Show, Data, Typeable)

fromPriority :: C'v4l2_priority -> Priority
35 toPriority :: Priority -> C'v4l2_priority
(fromPriority, toPriority) = (fromCEnum spec (PriorityUnknown . fromIntegral), ↯
    ↵ toCEnum spec isUnknown unUnknown) where
    spec =
        [ ( PriorityUnset      , c'V4L2_PRIORITY_UNSET      )
          , ( PriorityBackground , c'V4L2_PRIORITY_BACKGROUND )
40      , ( PriorityInteractive , c'V4L2_PRIORITY_INTERACTIVE )
          , ( PriorityDefault   , c'V4L2_PRIORITY_DEFAULT   )
          , ( PriorityRecord    , c'V4L2_PRIORITY_RECORD    )
        ]
    isUnknown (PriorityUnknown _) = True
45    isUnknown _ = False
    unUnknown (PriorityUnknown x) = fromIntegral x
    unUnknown _ = error "Graphics.V4L2.Priority.toPriority.unUnknown"

{- | Get priority. -}
50 getPriority :: Device -> IO Priority
getPriority d = fromPriority `fmap` ioctl d C'VIDIOC_G_PRIORITY

{- | Set priority. -}
setPriority :: Device -> Priority -> IO ()
55 setPriority d p = ioctl d C'VIDIOC_S_PRIORITY (toPriority p)

```

## 18 Graphics/V4L2/Types.hs

```

{- |
Module      : Graphics.V4L2.Types
Maintainer  : claude@mathr.co.uk
Stability   : no
5  Portability : no
-}
module Graphics.V4L2.Types
  ( module Graphics.V4L2.Types.Internal
  ) where
10
import Graphics.V4L2.Types.Internal
  ( Fraction(..)
  )

```

## 19 Graphics/V4L2/Types/Internal.hs

```

{-# LANGUAGE DeriveDataTypeable #-}
{- |
Module      : Graphics.V4L2.Types.Internal
Maintainer  : claude@mathr.co.uk
5  Stability   : no
Portability : no

```

```

-}
module Graphics.V4L2.Types.Internal
  ( Fraction(..)
10    , fromFraction
    ) where

import Data.Data (Data)
import Data.Word (Word32)
15 import Data.Typeable (Typeable)

import Bindings.Linux.VideoDev2

{- | Fraction type. -}
20 data Fraction = Fraction{ fractionNumerator, fractionDenominator :: Word32 }
    deriving (Eq, Ord, Read, Show, Data, Typeable)

{- | Unmarshal fraction. -}
fromFraction :: C'v4l2_fract -> Fraction
25 fromFraction f = Fraction
    { fractionNumerator = c'v4l2_fract 'numerator f
    , fractionDenominator = c'v4l2_fract 'denominator f
    }

```

## 20 Graphics/V4L2/VideoCapture.hs

```

{- |
Module      : Graphics.V4L2.VideoCapture
Maintainer  : claude@mathr.co.uk
Stability   : no
5  Portability : no
-}
module Graphics.V4L2.VideoCapture
  ( module Graphics.V4L2.VideoCapture.Read
    ) where
10 import Graphics.V4L2.VideoCapture.Read

```

## 21 Graphics/V4L2/VideoCapture/Read.hs

```

{- |
Module      : Graphics.V4L2.VideoCapture.Read
Maintainer  : claude@mathr.co.uk
Stability   : no
5  Portability : no

Video capture using Read I/O.
-}
module Graphics.V4L2.VideoCapture.Read
10  ( withFrame
    ) where

import Foreign (Ptr, allocaBytes)
import Foreign.C (throwErrnoIfMinus1)
15 import Bindings.LibV4L2 (c'v4l2_read)

import Graphics.V4L2.Device (Device)

```

```

import Graphics.V4L2.Format (ImageFormat, imageSize)

20 {- | Capture a video frame. -}
withFrame :: Device -> ImageFormat -> (Ptr a -> Int -> IO b) -> IO b
withFrame d f a = do
    allocaBytes (imageSize f) $ \p -> do
        n <- throwErrnoIfMinus1 "Graphics.V4L2.VideoCapture.Read.withFrame" $
25         c'v4l2_read (fromIntegral d) p (fromIntegral $ imageSize f)
        a p (fromIntegral n)

```

## 22 Graphics/V4L2/VideoInput.hs

```

{-# LANGUAGE DeriveDataTypeable, GeneralizedNewtypeDeriving #-}
{- |
Module      : Graphics.V4L2.VideoInput
Maintainer  : claude@mathr.co.uk
5  Stability : no
Portability : no
-}
module Graphics.V4L2.VideoInput
10  ( VideoInputID()
    , VideoInputInfo(..)
    , VideoInputType(..)
    , VideoInputStatus(..)
    , VideoInputCapability(..)
    , videoInputs
15  , getVideoInput
    , setVideoInput
    ) where

import Control.Exception as E (catch, throwIO)
20 import Data.Bits (testBit)
import Data.Data (Data)
import Data.Map (Map)
import qualified Data.Map as M
import Data.Set (Set)
25 import qualified Data.Set as S
import Data.Typeable (Typeable)
import Data.Word (Word32)
import GHC.IO.Exception (IOErrorType(InvalidArgument), ioe_type)

30 import Bindings.Linux.VideoDev2

import Foreign.Extra.CEnum (fromCEnum)
import Foreign.Extra.BitSet (fromBitSet)
import Foreign.Extra.String (fromString)
35 import Graphics.V4L2.Device (Device)
import Graphics.V4L2.IOctl (ioctl, ioctl_, ioctl', zero)
import Graphics.V4L2.VideoStandard (VideoStandard)
import Graphics.V4L2.VideoStandard.Internal (fromVideoStandard)

40
c'V4L2_IN_CAP_PRESETS :: Word32
c'V4L2_IN_CAP_PRESETS = 1
c'V4L2_IN_CAP_CUSTOM_TIMINGS :: Word32
c'V4L2_IN_CAP_CUSTOM_TIMINGS = 2
45 c'V4L2_IN_CAP_STD :: Word32

```

```

c'V4L2_IN_CAP_STD = 4
newtype AudioInputID = AudioInputID Int
    deriving (Eq, Ord, Enum, Bounded, Read, Show, Data, Typeable, Num, Integral, ↯
        ↪ Real)
newtype TunerID = TunerID Int
50    deriving (Eq, Ord, Enum, Bounded, Read, Show, Data, Typeable, Num, Integral, ↯
        ↪ Real)

{- | Video input index. -}
newtype VideoInputID = VideoInputID Int
    deriving (Eq, Ord, Enum, Bounded, Read, Show, Data, Typeable, Num, Integral, ↯
        ↪ Real)
55
{- | Video input info. -}
data VideoInputInfo = VideoInputInfo
    { videoInputName      :: String
    , videoInputType      :: VideoInputType
    , videoInputAudio     :: Set AudioInputID
    , videoInputTuner     :: Set TunerID
    , videoInputStandard  :: VideoStandard
    , videoInputStatus    :: Set VideoInputStatus
    , videoInputCapability :: Set VideoInputCapability
65    }
    deriving (Eq, Ord, Read, Show, Typeable)

{- | Video input type. -}
data VideoInputType
70    = TunerInput
    | CameraInput
    | UnknownInput Word32
    deriving (Eq, Ord, Read, Show, Data, Typeable)

75 fromVideoInputType :: Word32 -> VideoInputType
fromVideoInputType = fromCEnum
    [ ( TunerInput  , c'V4L2.INPUT.TYPE.TUNER  )
    , ( CameraInput , c'V4L2.INPUT.TYPE.CAMERA )
    ]    UnknownInput
80
{- | Video input status. -}
data VideoInputStatus
    = NoPower
    | NoSignal
    , NoColor
85    | HFlip
    | VFlip
    | NoHLock
    | ColorKill
    , NoSync
90    | NoEqu
    | NoCarrier
    | Macrovision
    | NoAccess
95    | Vtr
    | UnknownStatus Word32
    deriving (Eq, Ord, Read, Show, Data, Typeable)

fromVideoInputStatus :: Word32 -> Set VideoInputStatus

```

```

100 fromVideoInputStatus = fromBitSet
    [ ( NoPower      , c'V4L2_IN_ST_NO_POWER      )
    , ( NoSignal     , c'V4L2_IN_ST_NO_SIGNAL     )
    , ( NoColor      , c'V4L2_IN_ST_NO_COLOR      )
    , ( HFlip        , c'V4L2_IN_ST_HFLIP         )
105   , ( VFlip        , c'V4L2_IN_ST_VFLIP         )
    , ( NoHLock      , c'V4L2_IN_ST_NO_HLOCK      )
    , ( ColorKill    , c'V4L2_IN_ST_COLOR_KILL    )
    , ( NoSync       , c'V4L2_IN_ST_NO_SYNC       )
    , ( NoEqu        , c'V4L2_IN_ST_NO_EQU        )
110   , ( NoCarrier    , c'V4L2_IN_ST_NO_CARRIER   )
    , ( Macrovision  , c'V4L2_IN_ST_MACROVISION   )
    , ( NoAccess     , c'V4L2_IN_ST_NO_ACCESS     )
    , ( Vtr          , c'V4L2_IN_ST_VTR           )
    ] UnknownStatus

115 {- | Video input capabilities. -}
data VideoInputCapability
    = Presets
    | CustomTimings
120   | InputStd
    | UnknownCapability Word32
    deriving (Eq, Ord, Read, Show, Data, Typeable)

fromVideoInputCapability :: Word32 -> Set VideoInputCapability
125 fromVideoInputCapability = fromBitSet
    [ ( Presets      , c'V4L2_IN_CAP_PRESETS      )
    , ( CustomTimings , c'V4L2_IN_CAP_CUSTOM_TIMINGS )
    , ( InputStd      , c'V4L2_IN_CAP_STD          )
    ] UnknownCapability

130 {- | Enumerate video inputs. -}
videoInputs :: Device -> IO (Map VideoInputID VideoInputInfo)
videoInputs = enuminputs' 0

135 enuminputs' :: VideoInputID -> Device -> IO (Map VideoInputID VideoInputInfo)
enuminputs' n h = do
    mi <- (Just 'fmap' enuminput h n) 'E.catch' (\e -> case ioe_type e of
        InvalidArgument -> return Nothing
        _ -> throwIO e)
140   case mi of
        Just i -> M.insert n i 'fmap' enuminputs' (n + 1) h
        Nothing -> return M.empty

enuminput :: Device -> VideoInputID -> IO VideoInputInfo
145 enuminput h n = do
    i' <- ioctl h C'VIDIOC_ENUMINPUT ==<< return . (\s->s{ c'v4l2_input'index = \
        \< fromIntegral n }) ==<< zero
    return (decodeInput i')

decodeInput :: C'v4l2_input -> VideoInputInfo
150 decodeInput i = VideoInputInfo
    { videoInputName = fromString $ c'v4l2_input'name i
    , videoInputType = fromVideoInputType $ c'v4l2_input'type i
    , videoInputAudio = S.fromList [ fromIntegral ai | ai <- [0..31], c'v4l2_input \
        \< 'audioset i 'testBit' ai ]
    , videoInputTuner = S.fromList [ fromIntegral ti | ti <- [0..31], c'v4l2_input \

```

```

    ↵ 'tuner i 'testBit' ti ]
155   , videoInputStandard = fromVideoStandard $ c'v4l2_input'std i
    , videoInputStatus = fromVideoInputStatus $ c'v4l2_input'status i
    , videoInputCapability = fromVideoInputCapability $ c'v4l2_input'reserved i !! ↵
    ↵ 0
  }

160 {- | Query the current video input.

      Exceptions:

          * InvalidArgument - this device has no video inputs
165 -}
getVideoInput :: Device -> IO VideoInputID
getVideoInput h = fromIntegral 'fmap' ioctl' h C'VIDIOC_G_INPUT

170 {- | Select the current video input.

      Exceptions:

          * InvalidArgument - no video input with this index
175          * ResourceBusy - the video input cannot be switched now
-}
setVideoInput :: Device -> VideoInputID -> IO ()
setVideoInput h i = ioctl_ h C'VIDIOC_S_INPUT (fromIntegral i)

```

## 23 Graphics/V4L2/VideoStandard.hs

```

{- |
Module      : Graphics.V4L2.VideoStandard
Maintainer  : claude@mathr.co.uk
Stability   : no
5  Portability : no
-}
module Graphics.V4L2.VideoStandard
  ( module Graphics.V4L2.VideoStandard.Internal
  ) where
10
import Graphics.V4L2.VideoStandard.Internal hiding (fromVideoStandard)

```

## 24 Graphics/V4L2/VideoStandard/Internal.hs

```

{-# LANGUAGE DeriveDataTypeable, GeneralizedNewtypeDeriving #-}
{- |
Module      : Graphics.V4L2.VideoStandard.Internal
Maintainer  : claude@mathr.co.uk
5  Stability   : no
  Portability : no
-}
module Graphics.V4L2.VideoStandard.Internal
  ( VideoStandardID()
10   , VideoStandardInfo(..)
    , videoStandards
    , getVideoStandard
    , setVideoStandard
  )

```

```

    , detectVideoStandard
15  -- predefined standards
    , VideoStandard
    , VideoStandardType (..)
    , videoStandardPalB
    , videoStandardPalB1
20  , videoStandardPalG
    , videoStandardPalH
    , videoStandardPalI
    , videoStandardPalD
    , videoStandardPalD1
25  , videoStandardPalK
    , videoStandardPalM
    , videoStandardPalN
    , videoStandardPalNc
    , videoStandardPal60
30  , videoStandardNtscM
    , videoStandardNtscM_JP
    , videoStandardNtsc443
    , videoStandardNtscM_KR
    , videoStandardSecamB
35  , videoStandardSecamD
    , videoStandardSecamG
    , videoStandardSecamH
    , videoStandardSecamK
    , videoStandardSecamK1
40  , videoStandardSecamL
    , videoStandardSecamLC
    , videoStandardAtsc8Vsb
    , videoStandardAtsc16Vsb
    , videoStandardMn
45  , videoStandardB
    , videoStandardGh
    , videoStandardDk
    , videoStandardPalBg
    , videoStandardPalDk
50  , videoStandardPal
    , videoStandardNtsc
    , videoStandardSecamDk
    , videoStandardSecam
    , videoStandard525_60
55  , videoStandard625_50
    , videoStandardAtsc
    , videoStandardUnknown
    , videoStandardAll
    --
60  , fromVideoStandard
    ) where

import Control.Exception as E (catch, throwIO)
import Data.Data (Data)
65  import Data.Set (Set)
import Data.Map (Map)
import qualified Data.Map as M
import Data.Typeable (Typeable)
import Data.Word (Word64)
70  import GHC.IO.Exception (IOErrorType(InvalidArgument), ioe_type)

```

```

import Bindings.Linux.VideoDev2

import Foreign.Extra.BitSet (fromBitSet, toBitSet)
75 import Foreign.Extra.String (fromString)
import Graphics.V4L2.Device (Device)
import Graphics.V4L2.IOCtl (ioctl, ioctl_, ioctl', zero)
import Graphics.V4L2.Types (Fraction)
import Graphics.V4L2.Types.Internal (fromFraction)

80 {- | Video standard. -}
type VideoStandard = Set VideoStandardType

{- | Elementary video standard flags. -}
85 data VideoStandardType
    = StdPalB
    | StdPalB1
    | StdPalG
    | StdPalH
90 | StdPalI
    | StdPalD
    | StdPalD1
    | StdPalK
    | StdPalM
95 | StdPalN
    | StdPalNc
    | StdPal60
    | StdNtscM
    | StdNtscMJP
100 | StdNtsc443
    | StdNtscMKr
    | StdSecamB
    | StdSecamD
    | StdSecamG
105 | StdSecamH
    | StdSecamK
    | StdSecamK1
    | StdSecamL
    | StdSecamLC
110 | StdAtsc8Vsb
    | StdAtsc16Vsb
    | StdUnknown Word64
    deriving (Eq, Ord, Read, Show, Data, Typeable)

115 {- | internal -}
fromVideoStandard :: Word64 -> VideoStandard
{- | internal -}
toVideoStandard :: VideoStandard -> Word64
(fromVideoStandard, toVideoStandard) = (fromBitSet spec StdUnknown, toBitSet ↯
    ↵ spec isStdUnknown unStdUnknown) where
120 spec =
    [ ( StdPalB      , c'V4L2.STD.PAL.B      )
    , ( StdPalB1     , c'V4L2.STD.PAL.B1     )
    , ( StdPalG      , c'V4L2.STD.PAL.G      )
    , ( StdPalH      , c'V4L2.STD.PAL.H      )
125 , ( StdPalI      , c'V4L2.STD.PAL.I      )
    , ( StdPalD      , c'V4L2.STD.PAL.D      )

```

```

    , ( StdPalD1      , c'V4L2.STD.PAL_D1      )
    , ( StdPalK       , c'V4L2.STD.PAL_K       )
    , ( StdPalM       , c'V4L2.STD.PAL_M       )
130   , ( StdPalN       , c'V4L2.STD.PAL_N       )
    , ( StdPalNc      , c'V4L2.STD.PAL_Nc      )
    , ( StdPal60      , c'V4L2.STD.PAL_60      )
    , ( StdNtscM      , c'V4L2.STD.NTSC_M      )
    , ( StdNtscMJP    , c'V4L2.STD.NTSC_M_JP    )
135   , ( StdNtsc443   , c'V4L2.STD.NTSC_443   )
    , ( StdNtscMKr    , c'V4L2.STD.NTSC_M_KR    )
    , ( StdSecamB     , c'V4L2.STD.SECAM_B     )
    , ( StdSecamD     , c'V4L2.STD.SECAM_D     )
    , ( StdSecamG     , c'V4L2.STD.SECAM_G     )
140   , ( StdSecamH     , c'V4L2.STD.SECAM_H     )
    , ( StdSecamK     , c'V4L2.STD.SECAM_K     )
    , ( StdSecamK1    , c'V4L2.STD.SECAM_K1    )
    , ( StdSecamL     , c'V4L2.STD.SECAM_L     )
    , ( StdSecamLC    , c'V4L2.STD.SECAM_LC    )
145   , ( StdAtsc8Vsb  , c'V4L2.STD.ATSC_8_VSB  )
    , ( StdAtsc16Vsb , c'V4L2.STD.ATSC_16_VSB )
    ]
isStdUnknown (StdUnknown _) = True
isStdUnknown _ = False
150   unStdUnknown (StdUnknown x) = x
    unStdUnknown _ = error "Graphics.V4L2.Video.Standard.Internal.toVideoStandard.↵
        ↵ unUnknown"

{- | Video standard identifier. -}
newtype VideoStandardID = VideoStandardID Int
155   deriving (Eq, Ord, Enum, Bounded, Read, Show, Data, Typeable, Num, Integral, ↵
        ↵ Real)

{- | Video standard information. -}
data VideoStandardInfo = VideoStandardInfo
    { videoStandardStandard :: VideoStandard
160   , videoStandardName :: String
    , videoStandardFramePeriod :: Fraction
    , videoStandardFrameLines :: Int
    }
    deriving (Eq, Ord, Read, Show, Data, Typeable)
165

{- | Enumerate video standards.

    Drivers may enumerate a different set of standards after
    switching the video input or output.
170 -}
videoStandards :: Device -> IO (Map VideoStandardID VideoStandardInfo)
videoStandards = enumstds' 0

enumstds' :: VideoStandardID -> Device -> IO (Map VideoStandardID ↵
    ↵ VideoStandardInfo)
175   enumstds' n h = do
    mi <- (Just 'fmap' enumstd h n) 'E.catch' (\e -> case ioe.type e of
        InvalidArgument -> return Nothing
        _ -> throwIO e)
    case mi of
180   Just i -> M.insert n i 'fmap' enumstds' (n + 1) h

```

```

    Nothing -> return M.empty

enumstd :: Device -> VideoStandardID -> IO VideoStandardInfo
enumstd h n = do
185   i' <- ioctl h C'VIDIOC_ENUMSTD ==<< return . (\s->s{ c'v4l2_standard'index = \
        ↳ fromIntegral n }) ==<< zero
    return (decodeStandard i')

decodeStandard :: C'v4l2_standard -> VideoStandardInfo
decodeStandard i = VideoStandardInfo
190   { videoStandardStandard = fromVideoStandard $ c'v4l2_standard'id i
    , videoStandardName = fromString $ c'v4l2_standard'name i
    , videoStandardFramePeriod = fromFraction $ c'v4l2_standard'frameperiod i
    , videoStandardFrameLines = fromIntegral $ c'v4l2_standard'framelines i
    }
195   {- | Get the current video standard. -}
getVideoStandard :: Device -> IO VideoStandard
getVideoStandard d = do
    s <- ioctl' d C'VIDIOC_G_STD
200   return (fromVideoStandard s)

    {- | Set the current video standard. -}
setVideoStandard :: Device -> VideoStandard -> IO ()
setVideoStandard d s = do
205   ioctl_ d C'VIDIOC_S_STD (toVideoStandard s)

    {- | Detect the current video standard. -}
detectVideoStandard :: Device -> IO VideoStandard
detectVideoStandard d = do
210   s <- ioctl' d C'VIDIOC_QUERYSTD
    return (fromVideoStandard s)

-- Known standards.

215   videoStandardPalB    :: VideoStandard
    videoStandardPalB1   :: VideoStandard
    videoStandardPalG    :: VideoStandard
    videoStandardPalH    :: VideoStandard
    videoStandardPalI    :: VideoStandard
220   videoStandardPalD    :: VideoStandard
    videoStandardPalD1   :: VideoStandard
    videoStandardPalK    :: VideoStandard
    videoStandardPalM    :: VideoStandard
    videoStandardPalN    :: VideoStandard
225   videoStandardPalNc   :: VideoStandard
    videoStandardPal60   :: VideoStandard

    videoStandardNtscM    :: VideoStandard
    videoStandardNtscM_JP :: VideoStandard
230   videoStandardNtsc443 :: VideoStandard
    videoStandardNtscM_KR :: VideoStandard

    videoStandardSecamB   :: VideoStandard
    videoStandardSecamD   :: VideoStandard
235   videoStandardSecamG   :: VideoStandard
    videoStandardSecamH   :: VideoStandard

```

```

videoStandardSecamK    :: VideoStandard
videoStandardSecamK1   :: VideoStandard
videoStandardSecamL    :: VideoStandard
240 videoStandardSecamLC :: VideoStandard

videoStandardAtsc8Vsb  :: VideoStandard
videoStandardAtsc16Vsb :: VideoStandard

245 videoStandardMn      :: VideoStandard
videoStandardB         :: VideoStandard
videoStandardGh        :: VideoStandard
videoStandardDk        :: VideoStandard
videoStandardPalBg     :: VideoStandard
250 videoStandardPalDk   :: VideoStandard
videoStandardPal       :: VideoStandard
videoStandardNtsc      :: VideoStandard
videoStandardSecamDk   :: VideoStandard
videoStandardSecam     :: VideoStandard
255 videoStandard525_60 :: VideoStandard
videoStandard625_50    :: VideoStandard
videoStandardAtsc      :: VideoStandard
videoStandardUnknown   :: VideoStandard
videoStandardAll       :: VideoStandard
260
videoStandardPalB      = fromVideoStandard c'V4L2.STD_PAL_B
videoStandardPalB1     = fromVideoStandard c'V4L2.STD_PAL_B1
videoStandardPalG      = fromVideoStandard c'V4L2.STD_PAL_G
videoStandardPalH      = fromVideoStandard c'V4L2.STD_PAL_H
265 videoStandardPalI    = fromVideoStandard c'V4L2.STD_PAL_I
videoStandardPalD      = fromVideoStandard c'V4L2.STD_PAL_D
videoStandardPalD1     = fromVideoStandard c'V4L2.STD_PAL_D1
videoStandardPalK      = fromVideoStandard c'V4L2.STD_PAL_K
videoStandardPalM      = fromVideoStandard c'V4L2.STD_PAL_M
270 videoStandardPalN    = fromVideoStandard c'V4L2.STD_PAL_N
videoStandardPalNc     = fromVideoStandard c'V4L2.STD_PAL_Nc
videoStandardPal60     = fromVideoStandard c'V4L2.STD_PAL_60

videoStandardNtscM      = fromVideoStandard c'V4L2.STD_NTSC_M
275 videoStandardNtscM_JP = fromVideoStandard c'V4L2.STD_NTSC_M_JP
videoStandardNtsc443    = fromVideoStandard c'V4L2.STD_NTSC_443
videoStandardNtscM_KR   = fromVideoStandard c'V4L2.STD_NTSC_M_KR

videoStandardSecamB     = fromVideoStandard c'V4L2.STD_SECAM_B
280 videoStandardSecamD   = fromVideoStandard c'V4L2.STD_SECAM_D
videoStandardSecamG     = fromVideoStandard c'V4L2.STD_SECAM_G
videoStandardSecamH     = fromVideoStandard c'V4L2.STD_SECAM_H
videoStandardSecamK     = fromVideoStandard c'V4L2.STD_SECAM_K
videoStandardSecamK1    = fromVideoStandard c'V4L2.STD_SECAM_K1
285 videoStandardSecamL   = fromVideoStandard c'V4L2.STD_SECAM_L
videoStandardSecamLC    = fromVideoStandard c'V4L2.STD_SECAM_LC

videoStandardAtsc8Vsb   = fromVideoStandard c'V4L2.STD_ATSC_8_VSB
videoStandardAtsc16Vsb  = fromVideoStandard c'V4L2.STD_ATSC_16_VSB
290
-- derived standards
videoStandardMn         = fromVideoStandard c'V4L2.STD_MN
videoStandardB          = fromVideoStandard c'V4L2.STD_B

```

```

295 videoStandardGh      = fromVideoStandard c'V4L2.STD_GH
videoStandardDk       = fromVideoStandard c'V4L2.STD_DK
videoStandardPalBg    = fromVideoStandard c'V4L2.STD_PAL_BG
videoStandardPalDk    = fromVideoStandard c'V4L2.STD_PAL_DK
videoStandardPal      = fromVideoStandard c'V4L2.STD_PAL
videoStandardNtsc     = fromVideoStandard c'V4L2.STD_NTSC
300 videoStandardSecamDk = fromVideoStandard c'V4L2.STD_SECAM_DK
videoStandardSecam    = fromVideoStandard c'V4L2.STD_SECAM
videoStandard525_60   = fromVideoStandard c'V4L2.STD_525_60
videoStandard625_50   = fromVideoStandard c'V4L2.STD_625_50
videoStandardAtsc     = fromVideoStandard c'V4L2.STD_ATSC
305 videoStandardUnknown = fromVideoStandard c'V4L2.STD_UNKNOWN
videoStandardAll      = fromVideoStandard c'V4L2.STD_ALL

```

## 25 LICENSE

Copyright (c)2011, Claude Heiland-Allen

All rights reserved.

```

5  Redistribution and use in source and binary forms, with or without
   modification, are permitted provided that the following conditions are met:

   * Redistributions of source code must retain the above copyright
     notice, this list of conditions and the following disclaimer.
10  * Redistributions in binary form must reproduce the above
     copyright notice, this list of conditions and the following
     disclaimer in the documentation and/or other materials provided
     with the distribution.
15  * Neither the name of Claude Heiland-Allen nor the names of other
     contributors may be used to endorse or promote products derived
     from this software without specific prior written permission.

20  THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
   "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
   LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
   A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
   OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
25  SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
   LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
   DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
   THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
   (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
30  OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```

## 26 Setup.hs

```

import Distribution.Simple
main = defaultMain

```

## 27 TODO

```

Device          V4L2 close() - Close a V4L2 device
(IOctl)         V4L2 ioctl() - Program a V4L2 device

```

|    |               |                                                                                                                                                 |
|----|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
|    |               | ioctl VIDIOC_CROPCAP - Information about the video ↗<br>↳ cropping and scaling abilities                                                        |
|    |               | ioctl VIDIOC_DBG_G_CHIP_IDENT - Identify the chips on a TV ↗<br>↳ card                                                                          |
| 5  |               | ioctl VIDIOC_DBG_G_REGISTER, VIDIOC_DBG_S_REGISTER - Read ↗<br>↳ or write hardware registers                                                    |
|    |               | ioctl VIDIOC_DQEVENT - Dequeue event                                                                                                            |
|    |               | ioctl VIDIOC_ENCODER_CMD, VIDIOC_TRY_ENCODER_CMD - Execute ↗<br>↳ an encoder command                                                            |
|    |               | ioctl VIDIOC_ENUMAUDIO - Enumerate audio inputs                                                                                                 |
|    |               | ioctl VIDIOC_ENUMAUDOUT - Enumerate audio outputs                                                                                               |
| 10 |               | ioctl VIDIOC_ENUM_DV_PRESETS - Enumerate supported Digital ↗<br>↳ Video presets                                                                 |
|    | Format        | ioctl VIDIOC_ENUM_FMT - Enumerate image formats                                                                                                 |
|    | Format        | ioctl VIDIOC_ENUM_FRAMESIZES - Enumerate frame sizes                                                                                            |
|    | Format        | ioctl VIDIOC_ENUM_FRAMEINTERVALS - Enumerate frame ↗<br>↳ intervals                                                                             |
|    | VideoInput    | ioctl VIDIOC_ENUMINPUT - Enumerate video inputs                                                                                                 |
| 15 |               | ioctl VIDIOC_ENUMOUTPUT - Enumerate video outputs                                                                                               |
|    | VideoStandard | ioctl VIDIOC_ENUMSTD - Enumerate supported video standards                                                                                      |
|    |               | ioctl VIDIOC_G_AUDIO, VIDIOC_S_AUDIO - Query or select the ↗<br>↳ current audio input and its attributes                                        |
|    |               | ioctl VIDIOC_G_AUDOUT, VIDIOC_S_AUDOUT - Query or select ↗<br>↳ the current audio output                                                        |
|    |               | ioctl VIDIOC_G_CROP, VIDIOC_S_CROP - Get or set the ↗<br>↳ current cropping rectangle                                                           |
| 20 | Control       | ioctl VIDIOC_G_CTRL, VIDIOC_S_CTRL - Get or set the value ↗<br>↳ of a control                                                                   |
|    |               | ioctl VIDIOC_G_DV_PRESET, VIDIOC_S_DV_PRESET - Query or ↗<br>↳ select the DV preset of the current input or output                              |
|    |               | ioctl VIDIOC_G_DV_TIMINGS, VIDIOC_S_DV_TIMINGS - Get or ↗<br>↳ set custom DV timings for input or output                                        |
|    |               | ioctl VIDIOC_G_ENC_INDEX - Get meta data about a ↗<br>↳ compressed video stream                                                                 |
|    |               | ioctl VIDIOC_G_EXT_CTRLS, VIDIOC_S_EXT_CTRLS, ↗<br>↳ VIDIOC_TRY_EXT_CTRLS - Get or set the value of ↗<br>↳ several controls, try control values |
| 25 |               | ioctl VIDIOC_G_FBUF, VIDIOC_S_FBUF - Get or set frame ↗<br>↳ buffer overlay parameters                                                          |
|    | Format        | pix ioctl VIDIOC_G_FMT, VIDIOC_S_FMT, VIDIOC_TRY_FMT - Get or ↗<br>↳ set the data format, try a format                                          |
|    |               | pixmp ioctl VIDIOC_G_FMT, VIDIOC_S_FMT, VIDIOC_TRY_FMT - Get or ↗<br>↳ set the data format, try a format                                        |
|    |               | win ioctl VIDIOC_G_FMT, VIDIOC_S_FMT, VIDIOC_TRY_FMT - Get or ↗<br>↳ set the data format, try a format                                          |
|    |               | vbi ioctl VIDIOC_G_FMT, VIDIOC_S_FMT, VIDIOC_TRY_FMT - Get or ↗<br>↳ set the data format, try a format                                          |
| 30 |               | svbi ioctl VIDIOC_G_FMT, VIDIOC_S_FMT, VIDIOC_TRY_FMT - Get or ↗<br>↳ set the data format, try a format                                         |
|    |               | raw ioctl VIDIOC_G_FMT, VIDIOC_S_FMT, VIDIOC_TRY_FMT - Get or ↗<br>↳ set the data format, try a format                                          |
|    |               | ioctl VIDIOC_G_FREQUENCY, VIDIOC_S_FREQUENCY - Get or set ↗<br>↳ tuner or modulator radio frequency                                             |
|    | VideoInput    | ioctl VIDIOC_G_INPUT, VIDIOC_S_INPUT - Query or select the ↗<br>↳ current video input                                                           |
|    |               | ioctl VIDIOC_G_JPEGCOMP, VIDIOC_S_JPEGCOMP                                                                                                      |
| 35 |               | ioctl VIDIOC_G_MODULATOR, VIDIOC_S_MODULATOR - Get or set ↗                                                                                     |

|    |                                                                 |                                                              |
|----|-----------------------------------------------------------------|--------------------------------------------------------------|
|    |                                                                 | ↳ modulator attributes                                       |
|    |                                                                 | ioctl VIDIOC_G_OUTPUT, VIDIOC_S_OUTPUT - Query or select ↗   |
|    |                                                                 | ↳ the current video output                                   |
|    |                                                                 | ioctl VIDIOC_G_PARM, VIDIOC_S_PARM - Get or set streaming ↗  |
|    |                                                                 | ↳ parameters                                                 |
|    | Priority                                                        | ioctl VIDIOC_G_PRIORITY, VIDIOC_S_PRIORITY - Query or ↗      |
|    | ↳ request the access priority associated with a file descriptor |                                                              |
|    |                                                                 | ioctl VIDIOC_G_SLICED_VBI_CAP - Query sliced VBI ↗           |
|    |                                                                 | ↳ capabilities                                               |
| 40 | VideoStandard                                                   | ioctl VIDIOC_G_STD, VIDIOC_S_STD - Query or select the ↗     |
|    | ↳ video standard of the current input                           |                                                              |
|    |                                                                 | ioctl VIDIOC_G_TUNER, VIDIOC_S_TUNER - Get or set tuner ↗    |
|    |                                                                 | ↳ attributes                                                 |
|    |                                                                 | ioctl VIDIOC_LOG_STATUS - Log driver status information      |
|    |                                                                 | ioctl VIDIOC_OVERLAY - Start or stop video overlay           |
|    |                                                                 | ioctl VIDIOC_QBUF, VIDIOC_DQBUF - Exchange a buffer with ↗   |
|    |                                                                 | ↳ the driver                                                 |
| 45 |                                                                 | ioctl VIDIOC_QUERYBUF - Query the status of a buffer         |
|    | Capability                                                      | ioctl VIDIOC_QUERYCAP - Query device capabilities            |
|    | Control                                                         | ioctl VIDIOC_QUERYCTRL, VIDIOC_QUERYMENU - Enumerate ↗       |
|    | ↳ controls and menu control items                               |                                                              |
|    |                                                                 | ioctl VIDIOC_QUERY_DV_PRESET - Sense the DV preset ↗         |
|    |                                                                 | ↳ received by the current input                              |
|    | VideoStandard                                                   | ioctl VIDIOC_QUERYSTD - Sense the video standard received ↗  |
|    | ↳ by the current input                                          |                                                              |
| 50 |                                                                 | ioctl VIDIOC_REQBUFS - Initiate Memory Mapping or User ↗     |
|    |                                                                 | ↳ Pointer I/O                                                |
|    |                                                                 | ioctl VIDIOC_S_HW_FREQ_SEEK - Perform a hardware frequency ↗ |
|    |                                                                 | ↳ seek                                                       |
|    |                                                                 | ioctl VIDIOC_STREAMON, VIDIOC_STREAMOFF - Start or stop ↗    |
|    |                                                                 | ↳ streaming I/O                                              |
|    |                                                                 | ioctl VIDIOC_SUBDEV_ENUM_FRAME_INTERVAL - Enumerate frame ↗  |
|    |                                                                 | ↳ intervals                                                  |
|    |                                                                 | ioctl VIDIOC_SUBDEV_ENUM_FRAME_SIZE - Enumerate media bus ↗  |
|    |                                                                 | ↳ frame sizes                                                |
| 55 |                                                                 | ioctl VIDIOC_SUBDEV_ENUM_MBUS_CODE - Enumerate media bus ↗   |
|    |                                                                 | ↳ formats                                                    |
|    |                                                                 | ioctl VIDIOC_SUBDEV_G_CROP, VIDIOC_SUBDEV_S_CROP - Get or ↗  |
|    |                                                                 | ↳ set the crop rectangle on a subdev pad                     |
|    |                                                                 | ioctl VIDIOC_SUBDEV_G_FMT, VIDIOC_SUBDEV_S_FMT - Get or ↗    |
|    |                                                                 | ↳ set the data format on a subdev pad                        |
|    |                                                                 | ioctl VIDIOC_SUBDEV_G_FRAME_INTERVAL, ↗                      |
|    |                                                                 | ↳ VIDIOC_SUBDEV_S_FRAME_INTERVAL - Get or set the ↗          |
|    |                                                                 | ↳ frame interval on a subdev pad                             |
|    |                                                                 | ioctl VIDIOC_SUBSCRIBE_EVENT, VIDIOC_UNSUBSCRIBE_EVENT - ↗   |
|    |                                                                 | ↳ Subscribe or unsubscribe event                             |
| 60 |                                                                 | V4L2 mmap() - Map device memory into application address ↗   |
|    |                                                                 | ↳ space                                                      |
|    |                                                                 | V4L2 munmap() - Unmap device memory                          |
|    | Device                                                          | V4L2 open() - Open a V4L2 device                             |
|    |                                                                 | V4L2 poll() - Wait for some event on a file descriptor       |
|    | VideoCapture.Read                                               | V4L2 read() - Read from a V4L2 device                        |
| 65 |                                                                 | V4L2 select() - Synchronous I/O multiplexing                 |
|    |                                                                 | V4L2 write() - Write to a V4L2 device                        |

## 28 v4l2.cabal

---

```

Name:                v4l2
Version:             0.1.0.3
Synopsis:             interface to Video For Linux Two (V4L2)
Description:
5   Higher-level interface to V4L2.

Homepage:            http://code.mathr.co.uk/v4l2

License:             BSD3
10  License-file:     LICENSE
Author:              Claude Heiland-Allen
Maintainer:          claude@mathr.co.uk
Category:            Graphics

15  Build-type:       Simple
Cabal-version:       >=1.6

Extra-source-files:  TODO

20  Library
    Exposed-modules:
        Graphics.V4L2,
        Graphics.V4L2.Device,
        Graphics.V4L2.Capability,
25  Graphics.V4L2.Priority,
        Graphics.V4L2.Control,
        Graphics.V4L2.Field,
        Graphics.V4L2.PixelFormat,
        Graphics.V4L2.ColorSpace,
30  Graphics.V4L2.Format,
        Graphics.V4L2.VideoInput,
        Graphics.V4L2.VideoStandard,
        Graphics.V4L2.VideoCapture,
        Graphics.V4L2.VideoCapture.Read,
35  Graphics.V4L2.Types
    Other-modules:
        Foreign.Extra.BitSet,
        Foreign.Extra.CEnum,
        Foreign.Extra.String,
40  Graphics.V4L2.IOctl,
        Graphics.V4L2.Field.Internal,
        Graphics.V4L2.PixelFormat.Internal,
        Graphics.V4L2.ColorSpace.Internal,
        Graphics.V4L2.VideoStandard.Internal,
45  Graphics.V4L2.Types.Internal
    Build-depends:
        base >= 3 && < 5,
        containers >= 0.3 && < 0.6,
        bindings-DSL >= 1.0.7 && < 1.1,
50  bindings-libv4l2 >= 0.1 && < 0.2,
        bindings-linux-videodev2 >= 0.1 && < 0.2,
        bindings-posix >= 1.2.3 && < 1.3,
        ioctl >= 0.0 && < 0.1
    ghc-options: -Wall
55  build-tools: hsc2hs

source-repository head

```

```
    type:      git
    location:   http://code.mathr.co.uk/v4l2.git
60
source-repository this
    type:      git
    location:   http://code.mathr.co.uk/v4l2.git
    tag:        v0.1.0.3
```