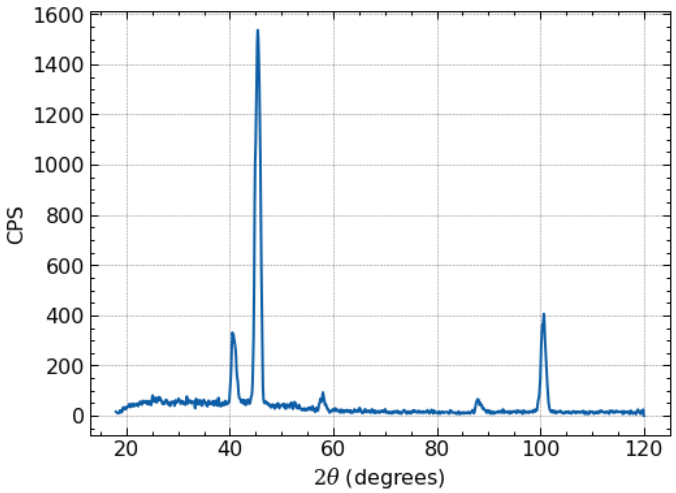
In [212]:
```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from scipy import constants as cnst


plt.rcParams['text.usetex'] = True
plt.style.use(['science','notebook','grid'])
#constants

c = cnst.c
Na = cnst.Avogadro
hbar = cnst.hbar
h=cnst.h
qe = cnst.e
me = cnst.m_e
pi = np.pi
```

LiF 20 kV (new)

In [213]:
```
data=pd.read_csv('data/lif20new.csv')
angle = data['angle'].values
countingrate = data['counting rate'].values

_ = plt.plot(angle, countingrate)#, marker = '.', linestyle = 'none')
_ = plt.xlabel(r'$2\theta$ (degrees)')
_ = plt.ylabel(r'CPS')
plt.savefig('lif20new', bbox_inches='tight')
plt.show()
```

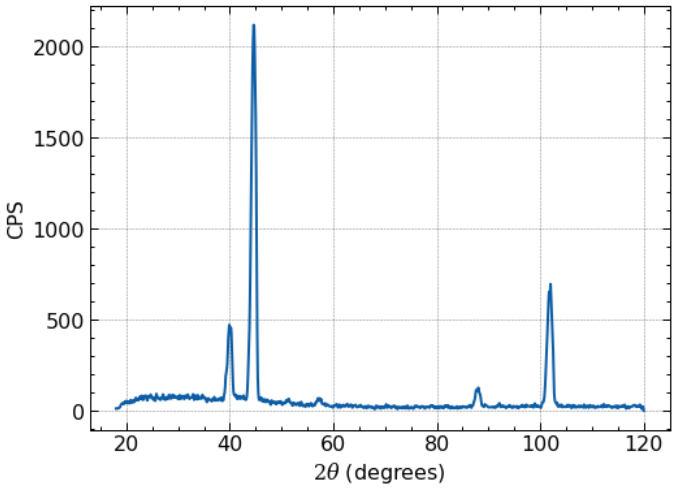


In [216]:
```
max1 = max(countingrate[0:241])
max2 = max(countingrate[242:281])
max3 = max(countingrate[432:812])
max4 = max(countingrate[812:])

index1 = np.where(countingrate == max1)
index2 = np.where(countingrate == max2)
index3 = np.where(countingrate == max3)
index4 = np.where(countingrate == max4)

theta1 = (angle[index1[0][0]])*(pi/180)
theta2 = (angle[index2[0][0]])*(pi/180)
theta3 = (angle[index3[0][0]])*(pi/180)
theta4 = (angle[index4[0][0]])*(pi/180)

rho = 2.635*(10**3)
M = 25.938*(10**(-3))

d = (M/(2*rho*Na))**(1/3)

wl_1 = 2*d*(np.sin(theta1/2)) #n=1
wl_2 = 2*d*(np.sin(theta2/2))

wl_3 = 2*d*(np.sin(theta3/2))/2 #n=2
wl_4 = 2*d*(np.sin(theta4/2))/2

wavelengths1 = np.array([wl_1, wl_2, wl_3, wl_4])
angles1 = np.array([theta1, theta2, theta3, theta4])*(180/pi)*0.5
print(wavelengths1)
print(angles1)
```

```
[1.39436986e-10 1.55466408e-10 1.39798805e-10 1.55092675e-10]
[20.25 22.7  43.95 50.35]
```

LiF 30 kV (new)

In [223]:
```
data=pd.read_csv('data/lif30new.csv')
angle = data['angle'].values
countingrate = data['counting rate'].values

_ = plt.plot(angle, countingrate)#, marker = '.', linestyle = 'none')
_ = plt.xlabel(r'$2\theta$ (degrees)')
_ = plt.ylabel(r'CPS')
plt.savefig('lif30new', bbox_inches='tight')
plt.show()
```



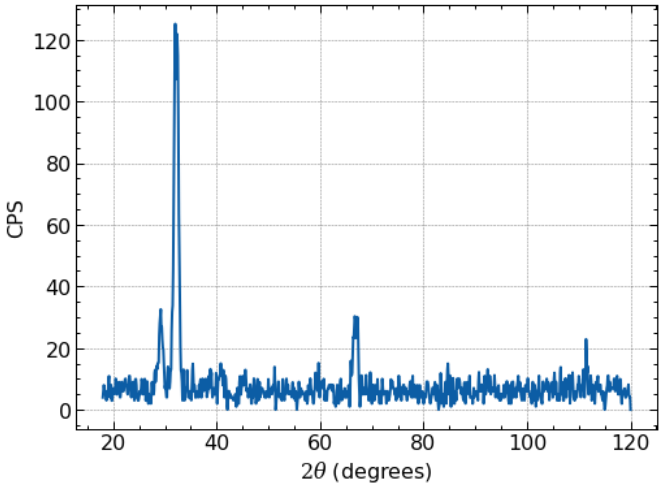
In [224]:
```
max1 = max(countingrate[0:241])
max2 = max(countingrate[242:281])
max3 = max(countingrate[432:812])
max4 = max(countingrate[812:])

index1 = np.where(countingrate == max1)
index2 = np.where(countingrate == max2)
index3 = np.where(countingrate == max3)
index4 = np.where(countingrate == max4)

theta1 = (angle[index1[0][0]])*(pi/180)
theta2 = (angle[index2[0][0]])*(pi/180)
theta3 = (angle[index3[0][0]])*(pi/180)
theta4 = (angle[index4[0][0]])*(pi/180)

rho = 2.635*(10**3)
M = 25.938*(10**(-3))

d = (M/(2*rho*Na))**(1/3)

wl_1 = 2*d*(np.sin(theta1/2)) #n=1
wl_2 = 2*d*(np.sin(theta2/2))

wl_3 = 2*d*(np.sin(theta3/2))/2 #n=2
wl_4 = 2*d*(np.sin(theta4/2))/2

wavelengths2 = np.array([wl_1, wl_2, wl_3, wl_4])
angles2 = np.array([theta1, theta2, theta3, theta4])*(180/pi)*0.5
print(wavelengths2)
print(angles2)
print('\n')
print(d)
```

```
[1.37456087e-10 1.52868003e-10 1.39925305e-10 1.56540813e-10]
[19.95 22.3  44.   51.  ]


2.014303880731949e-10
```

NaCl 20 kV (new)

In [225]:

```
data=pd.read_csv('data/nacl20new.csv')
angle = data['angle'].values
countingrate = data['counting rate'].values

_ = plt.plot(angle, countingrate)#, marker = '.', linestyle = 'none')
_ = plt.xlabel(r'$2\theta$ (degrees)')
_ = plt.ylabel(r'CPS')
#plt.axhline(y=35, color = 'red')
plt.savefig('nacl20new', bbox_inches='tight')
plt.show()
```



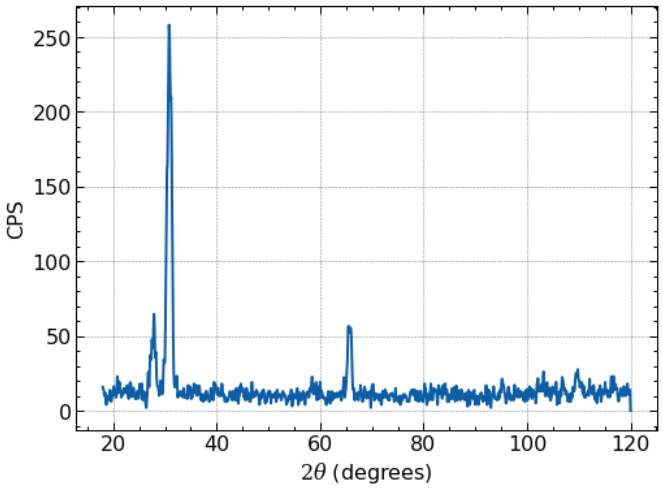
In [226]:

```
max1 = max(countingrate[0:123])
max2 = max(countingrate[123:154])
max3 = max(countingrate[412:424])
max4 = max(countingrate[424:500])
max5 = max(countingrate[500:823])
max6 = max(countingrate[823:])

index1 = np.where(countingrate == max1)
index2 = np.where(countingrate == max2)
index3 = np.where(countingrate == max3)
index4 = np.where(countingrate == max4)
index5 = np.where(countingrate == max5)
index6 = np.where(countingrate == max6)

theta1 = (angle[index1[0][0]])*(pi/180)
theta2 = (angle[index2[0][0]])*(pi/180)
theta3 = (angle[index3[0][0]])*(pi/180)
theta4 = (angle[index4[0][0]])*(pi/180)
theta5 = (angle[index5[0][0]])*(pi/180) #(angle[index5[0][0]])
theta6 = (angle[index6[0][0]])*(pi/180)

rho = 2.16*(10**3)
M = 58.46*(10**(-3))

d = (M/(2*rho*Na))**(1/3)

wl_1 = 2*d*(np.sin(theta1/2)) #n=1
wl_2 = 2*d*(np.sin(theta2/2))
wl_3 = 2*d*(np.sin(theta3/2))/2 #n=2
wl_4 = 2*d*(np.sin(theta4/2))/2
wl_5 = 2*d*(np.sin(theta5/2))/3 #n=3
wl_6 = 2*d*(np.sin(theta6/2))/3

wavelengths3 = np.array([wl_1, wl_2, wl_3, wl_4, wl_5, wl_6])
angles3 = np.array([theta1, theta2, theta3, theta4, theta5, theta6])*(180/pi)*0.5

print(wavelengths3)
print(angles3)
```

```
[1.42262828e-10 1.55564128e-10 1.40454647e-10 1.55134431e-10
 1.26732911e-10 1.55411035e-10]
[14.6  16.   29.85 33.35 42.35 55.7 ]
```

NaCl 30 kV (new)

In [227]:
```
data=pd.read_csv('data/nacl30new.csv')
angle = data['angle'].values
countingrate = data['counting rate'].values

_ = plt.plot(angle, countingrate)#, marker = '.', linestyle = 'none')
_ = plt.xlabel(r'$2\theta$ (degrees)')
_ = plt.ylabel(r'CPS')
plt.savefig('nacl30new', bbox_inches='tight')
plt.show()
```

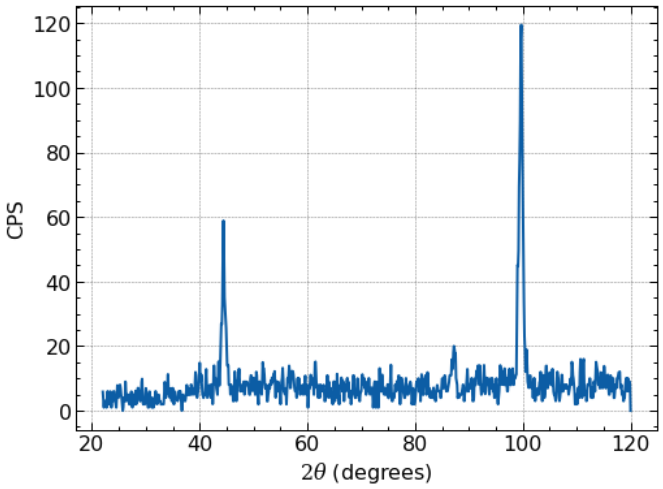


In [228]:
```
max1 = max(countingrate[0:108])
max2 = max(countingrate[123:138])
max3 = max(countingrate[412:424])
max4 = max(countingrate[424:500])
max5 = max(countingrate[500:823])
max6 = max(countingrate[823:])

index1 = np.where(countingrate == max1)
index2 = np.where(countingrate == max2)
index3 = np.where(countingrate == max3)
index4 = np.where(countingrate == max4)
index5 = np.where(countingrate == max5)
index6 = np.where(countingrate == max6)

theta1 = (angle[index1[0][0]])*(pi/180)
theta2 = (angle[index2[0][0]])*(pi/180)
theta3 = (angle[index3[0][0]])*(pi/180)
theta4 = (angle[index4[0][0]])*(pi/180)
theta5 = (angle[index5[0][0]])*(pi/180) #(angle[index5[0][0]])
theta6 = (angle[index6[0][0]])*(pi/180)

rho = 2.16*(10**3)
M = 58.46*(10**(-3))

d = (M/(2*rho*Na))**(1/3)

wl_1 = 2*d*(np.sin(theta1/2)) #n=1
wl_2 = 2*d*(np.sin(theta2/2))
wl_3 = 2*d*(np.sin(theta3/2))/2 #n=2
wl_4 = 2*d*(np.sin(theta4/2))/2
wl_5 = 2*d*(np.sin(theta5/2))/3 #n=3
wl_6 = 2*d*(np.sin(theta6/2))/3

wavelengths4 = np.array([wl_1, wl_2, wl_3, wl_4, wl_5, wl_6])
angles4 = np.array([theta1, theta2, theta3, theta4, theta5, theta6])*(180/pi)*0.5

print(wavelengths4)
print(angles4)
print('\n')
print(d)
```

```
[1.36057870e-10 1.49874481e-10 1.41094925e-10 1.52657504e-10
 1.38923067e-10 1.53915697e-10]
[13.95 15.4  30.   32.75 47.6  54.9 ]


2.8218984907688106e-10
```

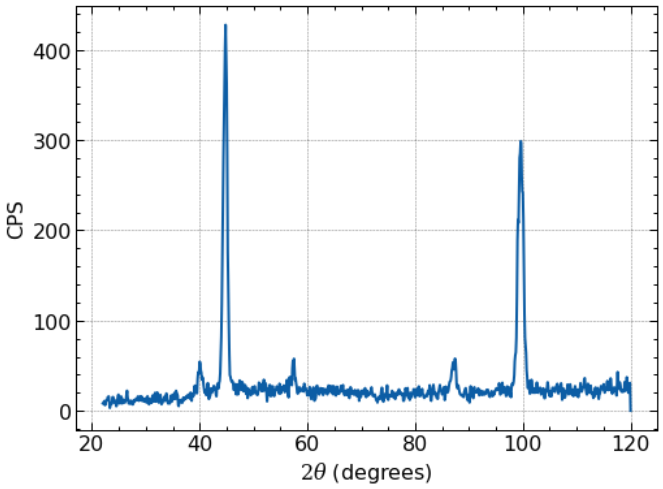
LiF 20 kV (old)

In [235]:
```
data=pd.read_csv('data/lif20old.csv')
angle = data['angle'].values
countingrate = data['counting rate'].values

_ = plt.plot(angle, countingrate)#, marker = '.', linestyle = 'none')
_ = plt.xlabel(r'$2\theta$ (degrees)')
_ = plt.ylabel(r'CPS')
plt.savefig('lif20old', bbox_inches='tight')
plt.show()
```



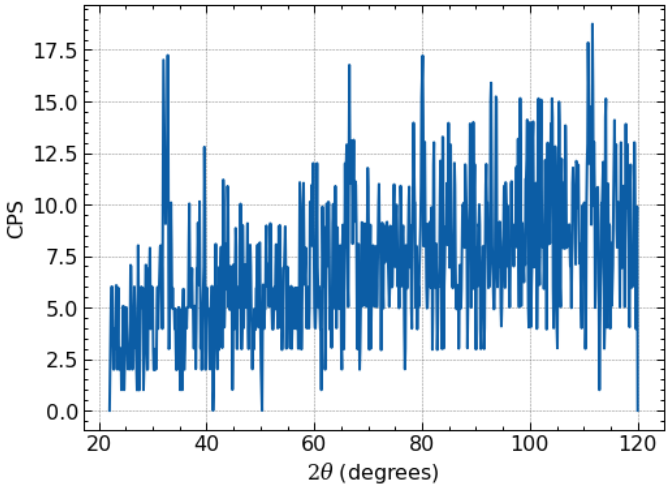
LiF 30 kV (old)

In [236]:
```
data=pd.read_csv('data/lif30old.csv')
angle = data['angle'].values
countingrate = data['counting rate'].values

_ = plt.plot(angle, countingrate)#, marker = '.', linestyle = 'none')
_ = plt.xlabel(r'$2\theta$ (degrees)')
_ = plt.ylabel(r'CPS')
plt.savefig('lif30old', bbox_inches='tight')
plt.show()
```



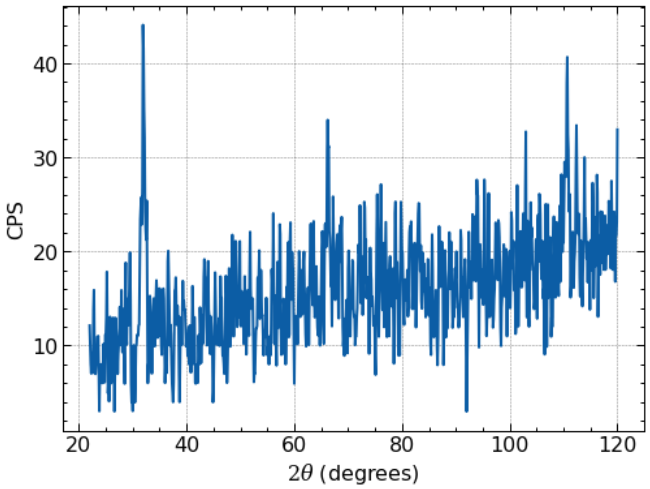
NaCl 20 kV (old)

In [242]:
```
data=pd.read_csv('data/nacl20old.csv')
angle = data['angle'].values
countingrate = data['counting rate'].values

_ = plt.plot(angle, countingrate)#, marker = '.', linestyle = 'none')
_ = plt.xlabel(r'$2\theta$ (degrees)')
_ = plt.ylabel(r'CPS')
plt.savefig('nacl20old', bbox_inches='tight')
plt.show()
```



NaCl 30 kV (old)

In [243]:
```
data=pd.read_csv('data/nacl30old.csv')
angle = data['angle'].values
countingrate = data['counting rate'].values

_ = plt.plot(angle, countingrate)#, marker = '.', linestyle = 'none')
_ = plt.xlabel(r'$2\theta$ (degrees)')
_ = plt.ylabel(r'CPS')
plt.savefig('nacl30old', bbox_inches='tight')
plt.show()
```

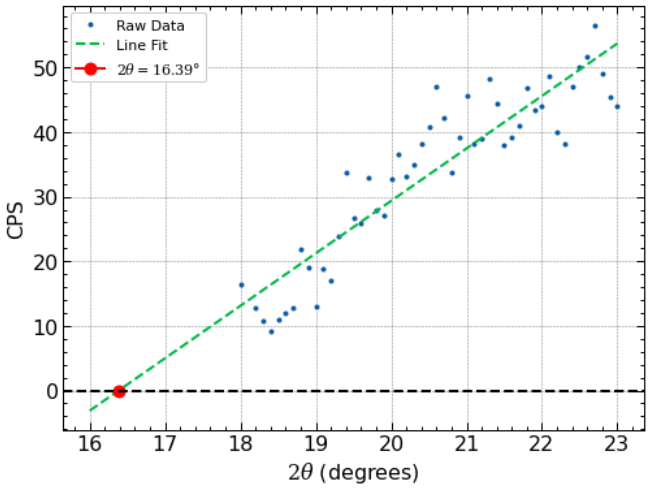


Planck's constant

Trial 1

In [423]:
```
data=pd.read_csv('data/lif20new.csv')

angle = data['angle'].values
countingrate = data['counting rate'].values

index = 50
xvals = angle[0:index]
yvals = countingrate[0:index]

a, b = np.polyfit(xvals, yvals, 1)

xfit = np.linspace(16, 23)
yfit = a*xfit + b

_ = plt.plot(xvals, yvals, marker = '.', linestyle = 'none', Label = 'Raw Data')
_ = plt.plot(xfit, yfit, linestyle = '--', label = 'Line Fit')
_ = plt.plot((-b/a), 0, marker = 'o', markersize = 9, color = 'red', label = r'$2\theta = 16.39°$')
_ = plt.xlabel(r'$2\theta$ (degrees)')
_ = plt.ylabel(r'CPS')
plt.axhline(y=0, color = 'black', linestyle = '--')
plt.legend(fontsize = 11, loc = 'upper left')
plt.savefig('lif20min', bbox_inches='tight')
plt.show()
```

```
<ipython-input-423-e1fad588d778>:15: MatplotlibDeprecationWarning: Case-insensitive properties were deprecated in 3.3 and support will be removed two minor releases later
  _ = plt.plot(xvals, yvals, marker = '.', linestyle = 'none', Label = 'Raw Data')
```



In [424]:
```
xroot = ((-b/a)*pi)/180
rho = 2.635*(10**3)
M = 25.938*(10**(-3))
V = 20*(10**3)

d = (M/(2*rho*Na))**(1/3)

lmin = 2*d*(np.sin(xroot/2))
print(xroot*(180/(2*pi)))
print(lmin)

h/((qe*V*lmin)/(c))
```

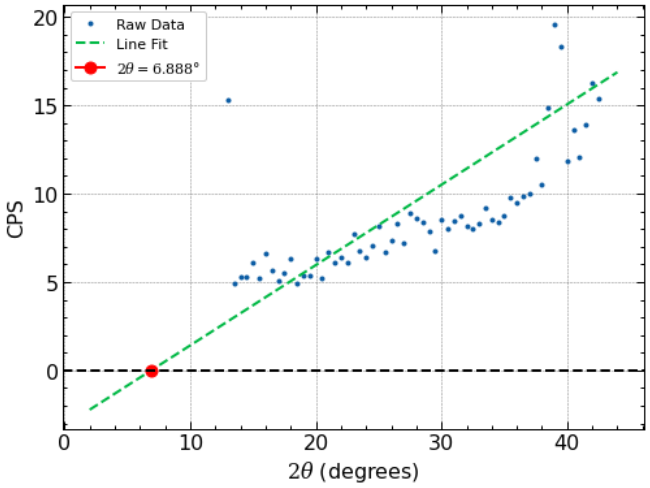
```
8.192933672844905
5.741042537416294e-11
```

Out[424]: 1.0798056069533868

Trial 2

In [425]:
```
data=pd.read_csv('data/lif30retrial.csv')

angle = data['angle'].values
countingrate = data['counting rate'].values

xvals = angle[0:61]
yvals = countingrate[0:61]

a, b = np.polyfit(xvals, yvals, 1)

xfit = np.linspace(2, 44)
yfit = a*xfit + b

_ = plt.plot(xvals[0:60], yvals[0:60], marker = '.', linestyle = 'none', Label = 'Raw Data')
_ = plt.plot(xfit, yfit, linestyle = '--', label = 'Line Fit')
_ = plt.plot((-b/a), 0, marker = 'o', markersize = 9, color = 'red', label = r'$2\theta = 6.888°$')
_ = plt.xlabel(r'$2\theta$ (degrees)')
_ = plt.ylabel(r'CPS')
plt.axhline(y=0, color = 'black', linestyle = '--')
plt.legend(fontsize = 11, loc = 'upper left')
plt.savefig('lif30min', bbox_inches='tight')
plt.show()
```

```
<ipython-input-425-1cff98495f93>:14: MatplotlibDeprecationWarning: Case-insensitive properties were deprecated in 3.3 and support will be removed two minor releases later
  _ = plt.plot(xvals[0:60], yvals[0:60], marker = '.', linestyle = 'none', Label = 'Raw Data')
```



In [426]:
```
xroot = ((-b/a)*pi)/180
rho = 2.635*(10**3)
M = 25.938*(10**(-3))
V = 30*(10**3)

d = (M/(2*rho*Na))**(1/3)

lmin = 2*d*(np.sin(xroot/2))
print(xroot*(180/(2*pi)))
print(lmin)

h/((qe*V*lmin)/(c))
```

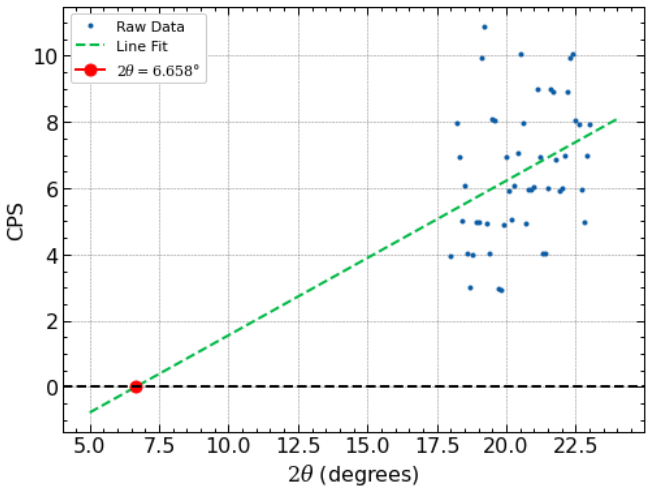
```
3.4441007152410337
2.4201741802277572e-11
```

Out[426]:
```
1.7076484197724475
```

Trial 3

In [427]:
```
data=pd.read_csv('data/nacl20new.csv')

angle = data['angle'].values
countingrate = data['counting rate'].values

index = 50
xvals = angle[0:index]
yvals = countingrate[0:index]

a, b = np.polyfit(xvals, yvals, 1)

xfit = np.linspace(5, 24)
yfit = a*xfit + b

_ = plt.plot(xvals, yvals, marker = '.', linestyle = 'none', Label = 'Raw Data')
_ = plt.plot(xfit, yfit, linestyle = '--', label = 'Line Fit')
_ = plt.plot((-b/a), 0, marker = 'o', markersize = 9, color = 'red', label = r'$2\theta = 6.658°$')
_ = plt.xlabel(r'$2\theta$ (degrees)')
_ = plt.ylabel(r'CPS')
plt.axhline(y=0, color = 'black', linestyle = '--')
plt.legend(fontsize = 11, loc = 'upper left')
plt.savefig('nacl20min', bbox_inches='tight')
plt.show()
```

```
<ipython-input-427-b41357723b8a>:15: MatplotlibDeprecationWarning: Case-insensitive properties were deprecated in 3.3 and support will be removed two minor releases later
  _ = plt.plot(xvals, yvals, marker = '.', linestyle = 'none', Label = 'Raw Data')
```



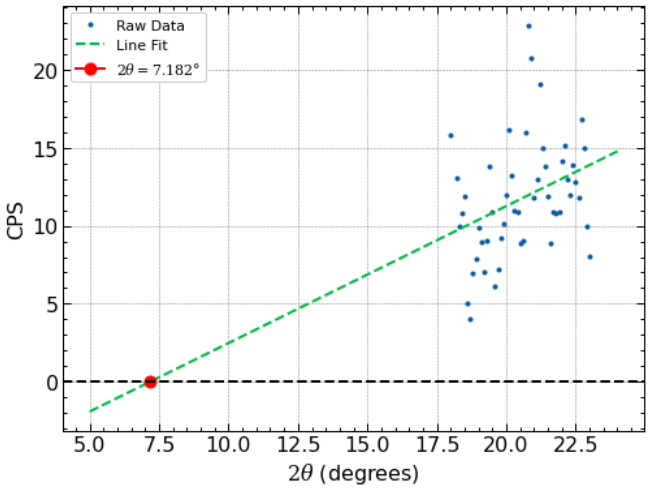
In [428]:
```
xroot = ((-b/a)*pi)/180
rho = 2.16*(10**3)
M = 58.46*(10**(-3))
V = 20*(10**3)

d = (M/(2*rho*Na))**(1/3)

lmin = 2*d*(np.sin(xroot/2))
print(xroot*(180/(2*pi)))
print(lmin)

h/((qe*V*lmin)/(c))
```

```
3.3287668848349905
3.2770856171523115e-11
```

Out[428]:
```
1.891683845308546
```

In [429]:
```
data=pd.read_csv('data/nacl30new.csv')

angle = data['angle'].values
countingrate = data['counting rate'].values

index = 50
xvals = angle[0:index]
yvals = countingrate[0:index]

a, b = np.polyfit(xvals, yvals, 1)

xfit = np.linspace(5, 24)
yfit = a*xfit + b

_ = plt.plot(xvals, yvals, marker = '.', linestyle = 'none', Label = 'Raw Data')
_ = plt.plot(xfit, yfit, linestyle = '--', label = 'Line Fit')
_ = plt.plot((-b/a), 0, marker = 'o', markersize = 9, color = 'red', label = r'$2\theta = 7.182°$')
_ = plt.xlabel(r'$2\theta$ (degrees)')
_ = plt.ylabel(r'CPS')
plt.axhline(y=0, color = 'black', linestyle = '--')
plt.legend(fontsize = 11, loc = 'upper left')
plt.savefig('nacl30min', bbox_inches='tight')
plt.show()
```

```
<ipython-input-429-f4d5e99fd6f7>:15: MatplotlibDeprecationWarning: Case-insensitive properties were deprecated in 3.3 and support will be remov
ed two minor releases later
  _ = plt.plot(xvals, yvals, marker = '.', linestyle = 'none', Label = 'Raw Data')
```



In [430]:
```
xroot = ((-b/a)*pi)/180
rho = 2.16*(10**3)
M = 58.46*(10**(-3))
V = 20*(10**3)

d = (M/(2*rho*Na))**(1/3)

lmin = 2*d*(np.sin(xroot/2))
print(xroot*(180/(2*pi)))
print(lmin)

h/((qe*V*lmin)/(c))
```

```
3.5911657573459714
3.535084595292091e-11
```

Out[430]:
```
1.7536242074421406
```

In [399]:
```
print(wavelengths1)
print(angles1)

print('\n')

print(wavelengths2)
print(angles2)
```

```
[1.39436986e-10 1.55466408e-10 1.39798805e-10 1.55092675e-10]
[20.25 22.7  43.95 50.35]


[1.37456087e-10 1.52868003e-10 1.39925305e-10 1.56540813e-10]
[19.95 22.3  44.   51.  ]
```

In [230]:
```
print(wavelengths3)
print(angles3)

print('\n')

print(wavelengths4)
print(angles4)
```

```
[1.42262828e-10 1.55564128e-10 1.40454647e-10 1.55134431e-10
 1.26732911e-10 1.55411035e-10]
[14.6  16.   29.85 33.35 42.35 55.7 ]


[1.36057870e-10 1.49874481e-10 1.41094925e-10 1.52657504e-10
 1.38923067e-10 1.53915697e-10]
[13.95 15.4  30.   32.75 47.6  54.9 ]
```

In [ ]:
```
```