Random Forrest: max depth hyperparameter: 20. Estimators: 100. R2 score: .96

Now, for modeling: I used grid search CV for choosing the number of estimators and max depth of the trees in the random forest.

In [57]:
```
rf = RandomForestRegressor()
parameters = {'n_estimators': [10, 20, 50, 100, 1000],
              'max_depth': [None, 1, 2, 3, 10, 20]}
grid_search = GridSearchCV(rf, parameters, cv=5, scoring='neg_mean_squared_error')
error_score='raise'
grid_search.fit(X_train, y_train.values.ravel())
```

Out[57]:

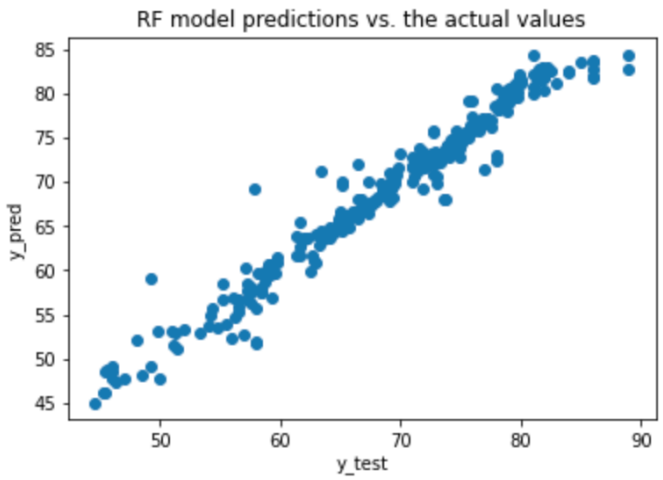
GridSearchCV
estimator: RandomForestRegressor
RandomForestRegressor
RandomForestRegressor()

In [58]:
```
# Make predictions using the trained model
y_pred = grid_search.predict(X_test)

# Print the best hyperparameters found
print(grid_search.best_params_)

from sklearn.metrics import r2_score
r2 = r2_score(y_test, y_pred)

print(f'R2 score: {r2:.2f}')
```

```
{'max_depth': 20, 'n_estimators': 100}
R2 score: 0.96
```

In [59]:
```
plt.scatter(y_test, y_pred)
plt.title("RF model predictions vs. the actual values")
plt.xlabel("y_test")
plt.ylabel("y_pred")
```

Out[59]:
```
Text(0, 0.5, 'y_pred')
```



We ended up with an R2 score of 0.96. The chart of predicted versus actual values looks good.

Ridge Regression: hyperparameter alpha 0.1, R2 0.96

Out[60]: `Text(0, 0.5, 'y_pred_2')`

Setting up CV grid search and getting our R2: also 0.96

In conclusion, the models, Ridge Multiple Linear Regression and Random Forest, when trained using cross validation to tune the hyperparameters of Alpha for Ridge, and the n_parameters and max_depth for Random Forest, actually gave the same R2 score of .96. This is pretty good, and both models give good predictions as shown by the scatter plots as well.